Maximising Influence Spread in Complex Networks by Utilising Community-Based Driver Nodes as Seeds

Abida Sadaf[1(✉)], Luke Mathieson[1], Piotr Bródka[2], and Katarzyna Musial[1]

[1] Complex Adaptive Systems Lab, School of Computer Science, University of Technology Sydney, Sydney, Australia
abida.sadaf@student.uts.edu.au
[2] Department of Artificial Intelligence, Wroclaw University of Science and Technology, Wroclaw, Poland
http://www.uts.edu.au, https://pwr.edu.pl/

Abstract. Finding a small subset of influential nodes to maximise influence spread in a complex network is an active area of research. Different methods have been proposed in the past to identify a set of seed nodes that can help achieve a faster spread of influence in the network. This paper combines driver node selection methods from the field of network control, with the divide-and-conquer approach of using community structure to guide the selection of candidate seed nodes from the driver nodes of the communities.

The use of driver nodes in communities as seed nodes is a comparatively new idea. We identify communities of synthetic (i.e., Random, Small-World and Scale-Free) networks as well as twenty-two real-world social networks. Driver nodes from those communities are then ranked according to a range of common centrality measures. We compare the influence spreading power of these seed sets to the results of selecting driver nodes at a global level. We show that in both synthetic and real networks, exploiting community structure enhances the power of the resulting seed sets.

Keywords: Influence · Complex Network · Social Networks · Seed Selection Methods · Driver Nodes · Communities

1 Introduction

Due to the prevailing use of online social networking sites, social networks are very much a hot topic in network science. Nowadays, we have a good understanding of network structures and attention has shifted more towards their prediction, influence, and control. Full control of social networks is very hard to achieve due to their varying structures, dynamics, and the complexities of human behaviour. This study looks into how driver nodes, which enable complex network control, can be used in the context of influence spread in the social network space. We use driver nodes at both the local and community level to 'divide and conquer' the

J. A. Lossio-Ventura et al. (Eds.): SIMBig 2022, CCIS 1837, pp. 126–141, 2023.
https://doi.org/10.1007/978-3-031-35445-8_10

time-consuming problem of driver node identification. Until recently, we did not know if and how the structure of social networks correlated with the number of driver nodes required to control the network [21]. As driver nodes play a key role in achieving control of a complex network, identifying them and studying their correlation with network structure measures can bring valuable insights, such as what network structures are easier to control, and how we can alter the structure in our favour to achieve the maximum control over the network. Our previous work [21] determines the relationship between some global network structure measures and the number of driver nodes. This study builds an understanding of how global network profiles of synthetic (random, small-world, scale-free) and real social networks influence the number of driver nodes needed for control. It focuses on global structural measures such as network density and how it can play an important role in determining the size of a suitable set of driver nodes. Our results show that as density increases in networks with structures exhibited by random, small world and scale free networks, the number of driver nodes tends to decrease. In this work we explore the potential that exploiting local structures (in this study we focus on communities) can offer in developing control of, and influencing, the network. Finding communities in a social network is itself a difficult task due to both dynamic and combinatorial factors [24].

This study explores the possibility of using community structure in social networks to reduce the cost of identifying driver nodes, and whether this remains a feasible approach for network control and influence spread methods.

Our main research questions for this work are stated as follows:

1. How can we rank driver nodes within communities to identify an optimal subset of driver nodes for use as seed nodes?
2. How quickly does influence spread from seed nodes chosen using driver node selection methods at the community level?
3. Does the percentage of influenced nodes increases or decreases when using driver node based seed selection methods in communities as compared to driver node based seed selection methods in the network as a whole, for both synthetic and real data?
4. How does the network structure (of synthetic or real networks) impact the percentage of nodes influenced with each method?

This paper contains the following sections: Sect. 2 describes related work and the main research challenge that is the focus of this study. Sections 3 and 4 describe (i) the research methodology in detail and (ii) include results and analysis of the experiments performed respectively. Finally, the conclusions drawn from the experiments and future work are discussed in Sect. 5.

2 Related Work

The Influence Maximisation problem aims at discovering an influential set of nodes that can influence the highest number of nodes in social networks in the shortest possible time. A set of these nodes can be used to propagate influence in terms of social media news, advertising, etc. Several algorithms have

been proposed to solve the influence maximisation problem that identify a set of nodes that is highly influential as compared to other nodes. For example Basic Greedy [13], CELF [14], CELF++ [10], Static Greedy [5], Nguyen's Method [20], Brog et al.'s Method [1], SKIM [6], TIM+ [26], IMM [25], Stop and Stare [18], Zohu et al.'s Method [28] and BCT [19] are some of those algorithms. Many algorithms have high run times when identifying a set of nodes to diffuse the influence through a social network, therefore there is a need to work on exploring different types of nodes if those can work towards achieving the high influence [12]. The problem of influence maximisation has high relevancy to the spreading of information on networks. The two most common network-based models are Independent Cascade model [13] and Threshold models [11]. In one of the previously proposed framework, the possible seed set has been identified by analysing the properties of the community structures in the networks. The CIM algorithm (i.e. Community-Based Influence Maximisation), utilises hierarchical clustering to detect communities from the networks and then uses the information of community structures to identify the possible seed nodes candidates, and at the end the final seed set is selected from the candidate seed nodes [4]. From the previous work such as [4] and [12], we can see, that by detecting communities and then selecting seed nodes from those communities can be an effective strategy to maximise influence.

From previous study [21], following main results were achieved, which are the basis for further new experiments in this current research work.

- Correlation between network density and number of driver nodes: For this purpose, network densities and number of driver nodes in those networks are plotted against each other to see the increase/decrease in number of driver nodes with the increase/decrease in the densities of the networks.
- Structural measures and density of driver nodes: In this step a comparison of structural measures like (Betweenness Centrality, Closeness Centrality, Nodes, Edges, Eigenvector Centrality and Clustering Coefficient) is presented with the density of number of driver nodes. Density of number of driver nodes is defined as total number of driver nodes divided by total number of nodes in the network.

In our proposed methods, we utilise driver nodes within the communities of networks for the influence spread using Linear Threshold Model. To make the driver nodes more influential, we propose different ranking mechanisms to see the number of nodes influenced after a certain time with a certain percentage of seed nodes in synthetic as well as real networks. The detail of network datasets has been presented in the later sections. We explain our method to select seed nodes from the communities in the next section.

3 Methodology

This work springs from the question, whether network control methods, in particular driver node selection, can be used to improve seed selection in influence models.

This prompts two possible approaches: (i) using driver nodes selected from the network as a whole, and (ii) using driver nodes selected at the community level as seeds. For all experiments, we used the Linear Threshold Model to model influence propagation. We used a set threshold of 0.5 for the network diffusion model. We have previously observed that a threshold value of at least 0.4 accelerates influence propagation [4].

3.1 Datasets Description

To enable comprehensive and robust testing of the proposed approaches, both generated and real-world social networks have been used. Following is a brief description of networks used in the experiments.

1. Generated Networks: we generated random, small-world and scale free networks from network size of (100, 200, 300, 400, 500) nodes. For each network size (from 100 to 500), we generated networks with increasing density, to the maximum density of 1. A total of 720 networks were generated [21].
2. Social Networks: we use 22 real-world social networks of varying size, the number of nodes and number of edges are presented in Table 2. The networks are available for download at SNAP[1].

3.2 Influence Spread Using Global Driver Nodes as Seeds

The first experiment focuses on the seed selection process from the global perspective. Driver nodes are selected from the network as a whole, ranked, and finally used as seeds in the influence process. The below described approach has been proposed in [22]. As it outperforms other state-of-the art ranking methods, it serves in this study as a benchmark to show a difference between global- and local-level seed selection methods. The steps are as follows:

1. Minimum Dominating Set method [17] has been used to identify the number of driver nodes from the networks. More detail of this process can be found in [21]. DMS has been found by using greedy algorithm. At start, the dominating set is empty. Then in each iteration of the algorithm, a vertex is added to the set such that it covers the maximum number of previously uncovered vertices. Then, if more than one vertex fulfils this criteria, the vertex is added randomly among the set of nominated vertices [23].
2. We ranked the nodes using different ranking mechanisms. The goal was to achieve an efficient set of nodes as seeds that can achieve maximum or full influence more quickly. The ranking mechanisms used are: Random, Degree Centrality, Closeness Centrality, Betweenness Centrality, Kempe Ranking, Degree-Closeness-Betweenness. We tested various seed set sizes: 1%, 10%, 20%, 30%, 40% and 50% of all detected driver nodes ranked these methods. In each of the methods, the driver nodes are ranked based on the following measures:

[1] http://snap.stanford.edu/.

- In Random (Driver Random – DR) we ranked the driver nodes randomly.
- In Degree seed selection (DD) we ranked the driver nodes based on their degree in descending order.
- For Closeness Centrality based seed selection method (Driver Closeness – DC), we ranked the nodes on the basis of their closeness centrality in descending order.
- For Betweenness Centrality based seed selection method (Driver Betweenness – DB), we ranked the nodes on the basis of their betweenness centrality in descending order.
- For Degree-Closeness-Betweenness method (Driver Degree Closeness Betweenness – DDCB), we ranked (in descending order) the driver nodes on the basis of the average of degree, closeness and betweenness centralities of each driver nodes.
- For Kempe ranking (Driver Kempe – DK), we start by spreading influence through all the driver nodes as seed nodes. So we calculate the total number of nodes influenced by each driver node already in the seed set, and then rank them in descending order. After ranking, we select a percentage of nodes that are required for a seed set.
- Linear Threshold Model (LTM) has been implemented for influence spread process. In LTM the idea is that a node becomes active if a sufficient part of its neighbourhood is active. Each node u has a threshold $t \in [0, 1]$. The threshold represents the fraction of neighbours of u that must be active in order for u to become active. At the beginning of the process, a small percentage of nodes (seeds) is set as active in order to start the process. In the next steps a node becomes active if the fraction of its active neighbours is greater than its threshold, and the whole process stops when no node is activated in the current step [7].

3.3 Influence Spread Using Local Driver Nodes as Seeds

The second experiment employs a new strategy: first identify communities in the network, and then identify driver nodes on a per-community basis.

Once driver nodes for each community are identified, they are then ranked using the same ranking mechanisms as in the first experiment, with seed sets chosen to cover all communities (detailed below). In detail, the approach is as follows:

1. Firstly, communities are identified in the network. This was done using Girvan-Newman algorithm [9]. The Girvan-Newman algorithm detects communities by progressively removing edges from the original graph in order of the highest betweenness centrality.
2. Within each community, candidate driver nodes were identified using the Minimum Dominating Set [17] approach as used with the whole network. Correlation between community densities and number of driver nodes is found by obtaining densities of the communities and identifying number of driver nodes in those communities by MDS method. Difference (Diff.) between total

number of driver nodes identified in overall networks (NDN) as compared to the number of driver nodes found in communities of those networks (NDNC) is also obtained. The Diff. tells us, the significance of identifying driver nodes within communities, like following a divide and conquer approach.

3. To rank the nodes, we introduce a multi-round selection process. This process effectively ranks driver nodes within each community according to the ranking criterion, then selects one node per community per round, in the order given by the ranking, until the total percentage to be chosen is reached. This is perhaps better explained by the following example, illustrated in Fig. 1. Consider a network with 1,000 nodes and 6 communities. Select a ranking method, in this case the node degree. Choose a target percentage of nodes to use as seed nodes, 1% in the example. Now, in order to choose 10 nodes from the driver nodes detected in the communities, we select 6 nodes at first – the highest degree node from each community, marked in yellow in the figure. In the second round, we can select at most 4 nodes to reach the target of 10 – from each community, we take the node with the second-highest node degree and rank these nodes according to their degrees and take the 4 nodes with the highest degree. We choose the same ranking mechanism for all the community based driver nodes seed selection methods i.e., the highest node degree, apart from the original ranking that is different in each technique as explained previously.
4. Influence spread in the overall network using Driver Based Seed Selection Methods is done by following a series of steps. Starting from identification of driver nodes from the networks, ranking of driver nodes based upon Random, Node Degree, Closeness Centrality, Betweenness Centrality, Kempe Ranking, Degree-Closeness-Betweenness Centralities combined. After ranking of driver nodes, we selected our seed set on the basis of percentage of nodes from that set. We run our LTM for different seed sets, namely for example 1%, 10%, 20%, 30%, 40% and 50%.
5. Influence spread through Driver Nodes in communities of Networks is done by identifying driver nodes in communities. However, there was a challenge of getting the ultimate seed set that has representation from all the communities of the network. For this purpose, we devised our ranking approach that makes sure that at least one driver node is selected from each community of the network to make sure that the nodes in those communities can also be part of the influence process. For each of the driver based seed selection methods, we used one unified approach to further rank the nodes so that we are able to select at least one node from each of the communities.

4 Results and Analysis

Six novel network level seed selection methods (i.e. Driver-Random (DR), Driver-Degree (DD), Driver-Closeness (DC), Driver-Betweenness (DB), Driver-Kempe (DK) and Driver-Degree-Closeness-Betweenness (DDCB)) have been proposed and tested on synthetic and real world networks before in [22] and

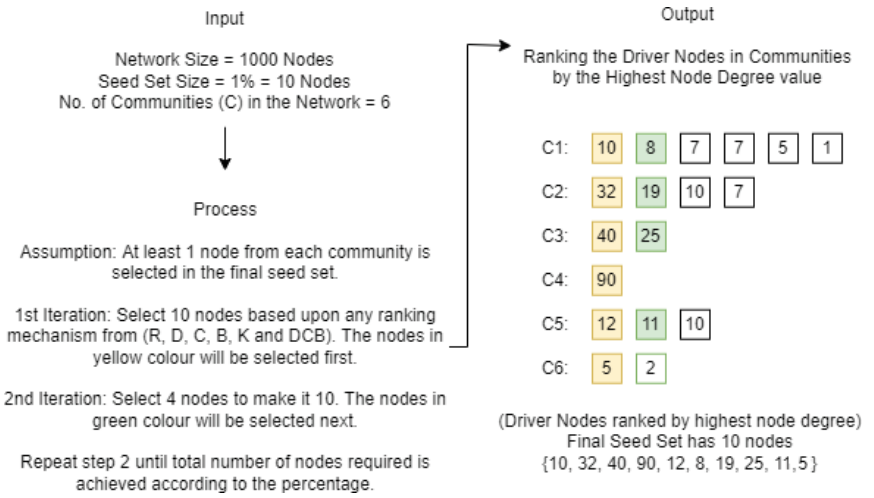

Fig. 1. An example showing the process for selecting seed nodes set from the driver nodes identified in network communities

the results show that those methods outperform their non-driver based counterparts. In this study, we use those methods but instead of selecting driver nodes from the global network, we propose a local approach where driver nodes are identified within the networks' communities. We name the new methods by adding C (for community) to the previously proposed methods (i.e., DRC - Driver-Random-Community, DDC - Driver-Degree-Community, DCC - Driver-Closeness-Community, DBC - Driver-Betweenness-Community, DKC - Driver-Kempe-Community and DDCBC - Driver-Degree-Closeness-Betweenness-Community). Below, we compare community based driver seed selection methods to network based driver seed selection methods.

4.1 Results from Generated Networks

This section covers the results and analysis of the experiments performed on generated networks.

What is the Speed and Reach of the Influence Spread? First, we compare the percentage of nodes influenced for global-level driver based seed selection methods and local-level (community) driver based seed selection methods. We perform the analysis iteration by iteration to see which seed selection methods enable to achieve the highest coverage the fastest.

In Fig. 2, we can see trend-lines for all the seed selection methods (when seed set size is 1% of all the driver nodes) for random, small-world and scale-free networks. DDCBC method outperforms other methods in almost all the experimented cases. We can see a 'head-start' in the trend-line of DDCBC (represented

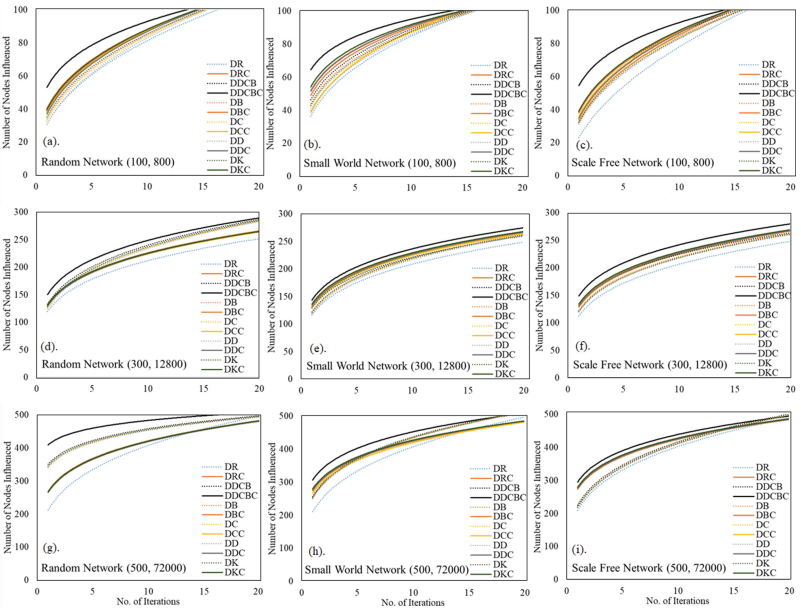

Fig. 2. Number of Nodes Influenced in Random, Small-World and Scale-Free Networks: when the number of nodes (N) is 100 and the number of edges (E) is 800 (Figures a, b and c); when N is 300 and E is 12800 (Figures d, e and f); when N is 500 and E is 72000 (Figures g, h and i). A Comparison of all methods for 20 iterations when the seed size is 1% is presented.

in black colour) for all the networks when number of nodes in the network is 100 and number of edges is 800. This means that in only few iterations, DDCBC enables to influence more nodes than in the case of other seed selection methods.

Results in Fig. 3 show that when the network is of small size, and density is approximately equal to 0.6, the influence spreads faster when using driver-community based seed selection methods than when the global-level driver based methods are employed. If we look at Fig. 3, the network of smaller densities (i.e. 0.4), where number of nodes is 300 and number of edges is 2,800, the difference between the global-level driver based methods and community-level driver based methods is not so big. But we do see a gap between DDCBC method and other methods. Which tells us that, so far, DDCBC ranking of driver nodes in communities is working better than when we are using driver nodes of communities as seed nodes.

Although the comparison is done on a very small size of seed set (1% of all driver nodes), in DDCBC, we still achieve more influence earlier in the spreading process when using community-level driver based methods. It also gives us another insight regarding larger networks, their structures and densities, and how those are connected to spreading influence. We see that the spread is faster when density is higher than 0.5 as in the case of networks presented in the Fig. 2

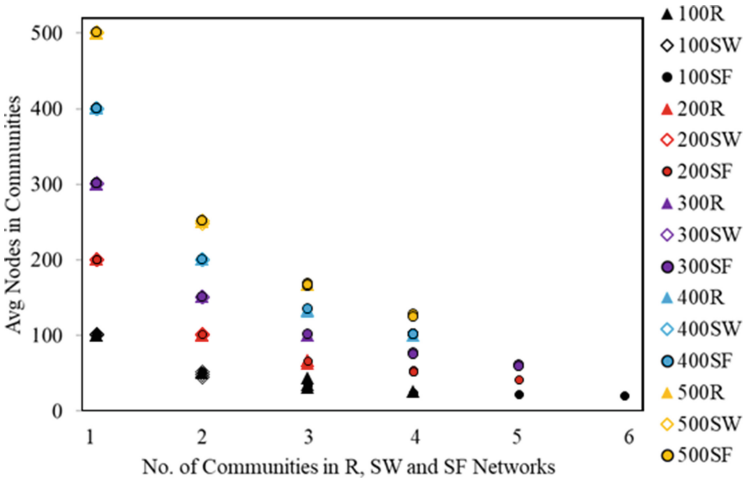

Fig. 3. Average Number of Nodes in Communities of Random, Small-World and Scale-Free Networks versus number of communities in those networks. Legend shows the Number of Nodes in communities of generated networks i.e. Random (R), Small-World (SW) and Scale-Free (SF).

(network with 500 nodes and 72,000 edges). We can see that in those cases, the driver-community based method DRC, DDC, DBC, DKC and DDCBC outperforms their counterpart methods DR, DD, DB, DK and DDCB.

Based upon these observations, we conclude it does not matter which type of network it is, as long as its density is higher than 0.5 it will respond to the community-based seed selection methods better and the spread will be faster. Also, regardless of the network density, community-based method – DDCBC – outperforms all other methods Fig. 2(a–f). This holds true for all the other settings as well. As when we have different edges for 100, 200, 300, 400 and 500 nodes networks.

How Much Advantage Do Community-Level Driver Based Seed Selection Methods Give? Given a number of iterations n and a method X, let $N_n^{infl}(X)$ denote the number of nodes influenced using the method X after n iterations. The Percentage Gain of method A over method B after n iterations is then given by:

$$\frac{N_n^{infl}(A) - N_n^{infl}(B)}{N} \times 100 \tag{1}$$

where N is the number of nodes in the network.

Table 1 shows the percentage gain of the DDCBC method over the global-level driver based methods. We represent only driver based methods (i.e. DR, DB, DC, DD, DK and DDCB), as the gain is higher over these methods as compared to other driver-community based methods (i.e. DRC, DBC, DCC, DDC and DKC) as well as they are our baseline for this study. Percentage gain

is calculated by knowing the maximum number of nodes influenced after 20 iterations when seed size is 1%.

From Table 1 we can see the maximum gain in when the average density of the communities of the network is greater than 0.5. When the density reaches 1 all the methods perform very similar as spread in fully connected network behaves in a very similar way regardless of applied seed selection method. This highlights our previous point that density of network plays an important part in how effective a network is going to respond to the influence spread process. We can see the highest gain for DDCBC method in random networks, but DDCBC outperforms all global-level driver based methods in all the networks, except for the networks with densities equal or very close to 1.

From Fig. 3, we can see the number of average nodes in communities versus the total number of communities in Random, Small-World and Scale-Free networks. The denser the network, the fewer communities we have, and those communities are denser than the previous ones. Hence, due to increase in community density, we see the higher percent gain in DDCBC method. The number of nodes influenced by DDCBC method increases, when there are fewer communities. Because when number of communities are less, they tend to be denser, hence the increase in number of nodes influenced. We see the difference in number of nodes influenced in DDCBC method which is bigger than compared to other methods.

4.2 Results from Social Networks

The observation that real-world social networks tend to contain dense communities suggests that community based driver node selection would have a significant advantage over global selection. This relationship with density is also apparent in the generated networks. To verify whether this intuition is correct, we conduct similar analysis to this performed on generated networks. First, we analyse the percentage of nodes influenced by each method over 100 iterations with a seed set size of 20% of driver nodes. We have run the experiments for the seed set sizes from 1%, 10%, 20%, 30%, 40% and 50%. We show the comparison in case of 20% seed size, as it is the lowest seed set level to reach maximum influence in at most 100 iterations. We note however that there is also improvements at smaller seed set sizes.

What is the Speed and Reach of the Influence Spread? Figures 4, 5 and 6 show a comparison between global-level driver based seed selection methods and community-level driver based seed selection methods. We grouped the networks on the basis of their sizes and densities to analyse the results effectively. From Fig. 4, we see a higher density of networks. The densities of these networks are: FB (0.01), Z (0.13), LC (0.003), LF (0.003), PF (0.007), FbG (0.003), FbP (0.002), FbPF (0.001) and FbT (0.002). Overall comparison tells us that, in these networks, there is less difference between the percentage of number of nodes influenced after 100 iterations. Which indicates that when the network's

Table 1. A percentage gain table shows the percentage gain of DDCBC method over other seed selection methods in influencing the nodes in Random, Small-World and Scale-Free networks when the seed set size is 1% after 20 iterations. N is number of nodes, E is number of edges, C is number of communities and CD is average community density.

N	E	C	CD	Random Networks						Small-World Networks						Scale-Free Networks					
			Avg ± SD	DR	DB	DC	DD	DK	DDCB	DR	DB	DC	DD	DK	DDCB	DR	DB	DC	DD	DK	DDCB
100	800	6	0.160.01	2	2	2	2	2	1	3	2	3	3	3	2	4	2	2	3	3	2
	1600	5	0.3±0.03	3	2	3	3	2	2	3	2	2	2	2	2	4	2	3	3	3	2
	2400	4	0.44±0.06	3	2	2	3	2	2	3	2	3	3	2	2	4	2	2	3	3	2
	3200	3	0.58±0.12	2	2	2	2	2	1	4	3	3	3	3	3	3	2	2	2	2	1
	4000	2	0.73±0.14	4	2	3	3	2	2	3	2	2	2	2	2	4	3	3	3	3	2
	4800	1	0.88±0.15	2	1	1	2	1	1	0	0	1	1	0	0	2	1	1	1	1	1
	4950	1	0.96±0.07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
200	2400	5	0.12±0.01	4	4	4	5	4	4	5	5	5	5	5	4	5	4	4	4	4	4
	4800	4	0.23±0.02	3	2	2	3	2	2	3	2	2	3	2	2	5	4	4	4	4	3
	7200	4	0.36±0.01	8	7	7	7	7	6	8	7	7	7	7	7	9	8	8	8	8	8
	9600	4	0.48±0.02	6	6	6	6	6	5	6	5	6	6	6	5	7	6	6	6	6	5
	12000	3	0.56±0.07	6	5	5	5	5	4	7	7	7	7	7	6	7	6	6	6	6	6
	14400	2	0.67±0.09	3	3	3	3	3	2	4	3	4	4	3	3	4	3	3	3	3	2
	16800	1	0.78±0.11	2	1	1	1	1	0	2	1	2	2	2	1	3	1	1	2	1	1
	19200	1	0.9±0.1	1	0	0	0	0	0	2	0	1	1	0	0	1	1	1	1	1	0
	19900	1	0.97±0.06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
300	12800	5	0.31±0.03	4	3	3	3	3	2	4	3	3	3	3	2	5	3	3	4	4	3
	19200	5	0.41±0.03	4	3	3	3	2	2	3	2	3	3	3	2	4	3	3	3	3	3
	22400	4	0.46±0.06	4	3	3	3	2	2	4	3	3	3	3	2	5	3	3	4	3	3
	25600	4	0.53±0.08	4	2	2	3	2	2	3	2	3	3	3	2	4	3	3	3	3	2
	28800	3	0.58±0.1	3	2	2	3	2	2	2	2	2	2	2	1	3	2	2	2	2	2
	32000	2	0.63±0.17	6	4	4	4	4	3	4	3	3	3	3	3	5	4	4	4	4	3
	35200	1	0.69±0.16	10	8	8	8	8	8	5	5	5	5	5	4	6	5	5	5	5	4
	38400	1	0.76±0.17	13	6	7	7	7	7	10	3	3	3	3	2	13	4	4	4	4	4
	41600	1	0.83±0.17	0	0	0	0	0	0	0	0	0	0	0	1	0	2	2	2	2	1
	44850	1	0.91±0.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
400	40000	4	0.43±0.12	23	21	21	22	21	21	5	4	4	4	4	4	5	4	4	4	4	3
	44000	4	0.48±0.12	25	22	22	22	22	22	4	4	4	4	4	3	6	4	4	5	4	4
	48000	4	0.53±0.12	25	21	21	21	21	21	9	8	8	8	8	7	10	8	8	9	8	8
	52000	4	0.58±0.12	22	12	12	13	12	12	12	11	11	11	11	11	13	11	12	12	12	11
	60000	3	0.67±0.14	18	12	12	12	12	12	10	9	9	10	10	9	12	10	10	11	11	10
	64000	2	0.76±0.07	13	8	9	9	8	9	7	7	7	7	7	6	8	7	7	7	7	7
	68000	1	0.83±0.03	8	6	6	6	6	6	5	4	4	4	4	4	7	5	6	6	6	5
	72000	1	0.88±0.03	4	1	1	2	1	1	5	4	4	4	4	4	4	3	3	3	3	3
	76000	1	0.93±0.03	1	0	0	0	0	0	1	1	1	1	1	0	1	2	2	3	2	2
	98000	1	0.98±0.03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
500	72000	4	0.52±0.1	23	15	15	16	16	15	11	6	6	6	6	6	12	11	11	11	11	10
	76800	3	0.56±0.1	21	16	16	16	16	16	11	6	7	7	6	6	7	6	6	6	6	6
	81600	4	0.6±0.09	19	13	14	14	13	13	10	7	8	8	8	7	8	7	7	7	7	6
	86400	3	0.69±0.01	19	13	13	13	13	13	10	2	2	2	2	1	9	8	8	8	8	7
	91200	3	0.73±0.01	15	14	14	14	14	14	7	3	3	3	3	3	3	2	2	2	2	1
	96000	3	0.76±0.01	12	10	10	10	10	10	7	2	2	2	2	1	5	3	3	4	4	3
	100800	1	0.81±0.01	8	8	8	9	9	8	7	1	1	2	1	1	4	3	3	3	3	2
	105200	1	0.84±0	3	4	5	5	5	5	3	0	1	1	0	0	2	1	1	1	1	0
	110000	2	0.88±0	0	3	3	4	4	3	0	0	0	0	0	0	0	1	1	1	1	0
	124750	2	0.97±0.06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

densities are higher, then there is more chance that seed selection methods are able to achieve influence faster. If we look at the Fb network in Fig. 4, its network density is 0.01 which is greater than the rest of the networks except the network

Table 2. A percentage gain table shows the percentage gain of DDCBC method over other seed selection methods in influencing the nodes of the social networks. Average Community Densities of the networks are as follows: FB (0.06 ± 0.02), ZKC (0.32 ± 0.4), Twitter (0.00029 ± 0.05), Diggs (0.00008 ± 0.007), Youtube (0.000012 ± 0.04), Ego (0.00034 ± 0.05), LC (0.007 ± 0.032), LF (0.0073 ± 0.09), PF (0.015 ± 0.54), MFb (0.001 ± 0.43), DHR (0.00085 ± 0.21), DRO (0.0005 ± 0.4), DHU (0.0004 ± 0.63), MG (0.0011 ± 0.03), L (0.0019 ± 0.54), FbAR (0.0014 ± 0.03), FbA (0.0015 ± 0.09), FbG (0.0075 ± 0.05), FbN (0.0013 ± 0.003), FbP (0.0049 ± 0.003), FbPF (0.004 ± 0.032) and Fbt (0.0051 ± 0.05)

N	E	C	Networks	Seed Selection Methods (20% of all nodes)										
				DR	DD	DC	DB	DDCB	DK	DRC	DDC	DCC	DBC	DKC
4039	88234	180	FB	28.68	25.03	24.94	25.94	25.15	24.59	21.59	21.59	22.19	22.28	21.14
34	78	2	ZKC	12.18	4.00	2.82	2.09	1.95	1.73	1.18	1.18	1.27	1.00	1
23371	32832	350	Twitter	37.81	27.83	26.80	26.78	20.16	26.77	23.81	23.80	23.74	23.06	21.22
1924000	3298475	156432	Diggs	42.49	39.05	36.76	36.47	38.37	39.21	20.11	18.89	17.67	16.53	19.85
1134891	2987625	54983	Youtube	42.00	38.02	35.12	32.79	32.59	33.92	3.51	2.71	1.91	1.11	6.45
23629	39195	75	Ego	24.83	15.34	14.33	14.33	17.15	21.81	9.64	10.62	11.14	9.05	8.89
4658	33116	517	LC	33.84	26.62	25.61	25.61	25.52	31.81	22.40	23.23	23.98	21.65	22.06
874	1309	97	LF	19.29	10.62	9.56	9.34	9.25	9.33	8.38	9.35	10.20	7.86	9.11
1858	12534	206	PF	10.62	6.66	5.43	5.21	5.13	5.25	2.94	3.78	4.64	2.60	2.71
22470	171002	2643	MFb	25.44	22.16	21.11	21.11	21.10	21.11	15.07	15.80	22.70	20.43	16.8
54574	498202	6420	DHR	39.77	35.42	33.21	32.00	31.90	34.2	6.78	7.26	7.73	5.21	6.01
41774	125826	4914	DRO	42.43	35.74	36.42	34.22	34.12	34.45	13.50	13.40	13.13	12.94	34.18
47539	222887	5592	DHU	45.40	35.77	34.52	34.33	34.13	38.85	26.35	27.02	25.84	25.61	25.33
37700	289003	4435	MG	30.54	27.43	26.07	26.25	26.07	26.34	16.14	15.49	16.05	10.35	14.86
7624	27806	759	L	26.55	25.25	24.04	23.82	23.79	23.81	18.34	18.11	17.75	17.71	17.70
50516	819306	5943	FbAR	39.97	32.40	31.18	30.95	30.93	31.30	29.43	29.14	30.85	28.56	29.28
13867	86858	1383	FbA	47.29	32.45	31.05	30.55	40.87	45.89	32.28	31.83	33.28	30.46	32.01
7058	89455	784	FbG	21.95	20.22	18.93	18.71	19.18	19.20	13.97	13.75	15.39	13.13	13.68
27918	206259	3284	FbN	33.82	23.03	22.00	21.95	21.96	22.01	12.85	12.64	12.18	12.10	12.40
5909	41729	562	FbP	31.73	22.90	21.76	21.40	21.87	21.89	15.89	15.47	15.15	14.90	15.31
11566	67114	1051	FbPF	39.61	32.21	30.85	30.57	30.39	30.48	26.30	26.12	25.85	25.21	26.21
3893	17262	387	FbT	25.93	22.84	24.70	24.29	17.73	17.77	19.37	18.46	13.71	13.36	17.63

Z which has the highest density of 0.14. If we compare the plots, we see that DDCBC method also works exceptionally better in most networks as compared to the rest of the methods. From Fig. 5, we see the networks with densities ranging from 0.0001 to 0.0009. Densities of these networks are: MFb (0.0006), DHR (0.0003), DRO (0.0001), DHU (0.0001), MG (0.0004), L (0.0009), FbAR (0.0006) and FbA (0.0009). With the lower density networks, we can see that the gain in driver community based methods is more prominent as compared to driver based methods. It means density of the network does play an important role to determine the total number of nodes influenced. From Fig. 6, we see the networks with the lowest densities ranging from 0.000002 to 0.0001. Densities of these networks are: Youtube (0.000004), Twitter (0.00012), Diggs (0.000002) and Ego (0.00014). In these networks, we see a huge gap between DDCBC method and the rest of the methods. Which means, even in the lowest density networks, when we locally construct communities, the density tend to increase as we can see from Table 2. Average community density of Youtube was calculated to be

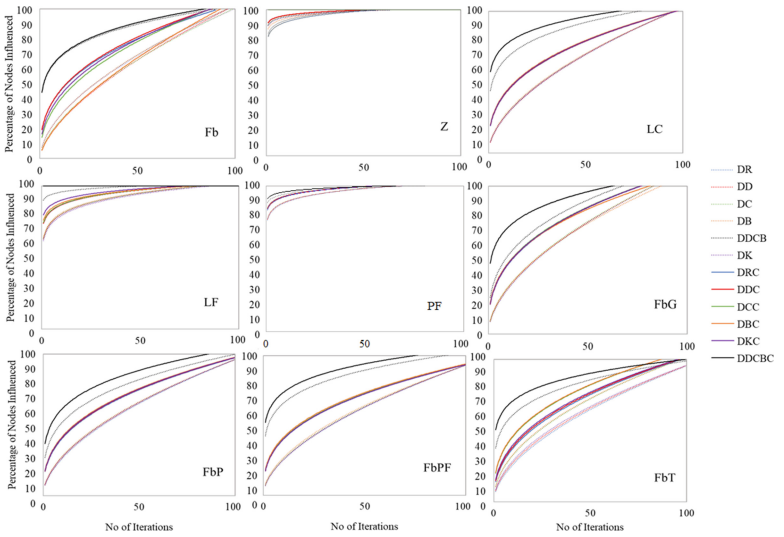

Fig. 4. Percentage of Number of Nodes Influenced in FB, Z, LC, LF, PF, FbG, FbP, FbPF and FbT Networks. A Comparison of all methods for 100 iterations.

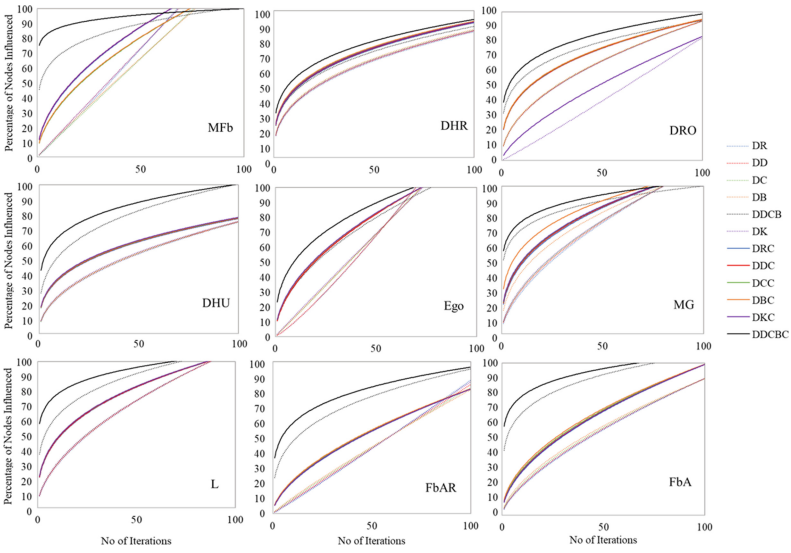

Fig. 5. Percentage of Number of Nodes Influenced in MFb, DHR, DRO, DHU, MG, L, FbAR and FbA Networks. A Comparison of all methods for 100 iterations.

0.000012 ± 0.04, which means if we compare it to the overall network density of 0.000004, it is notably denser. That is why, even in these networks, driver-community based methods specially DDCBC method outperforms the driver based methods.

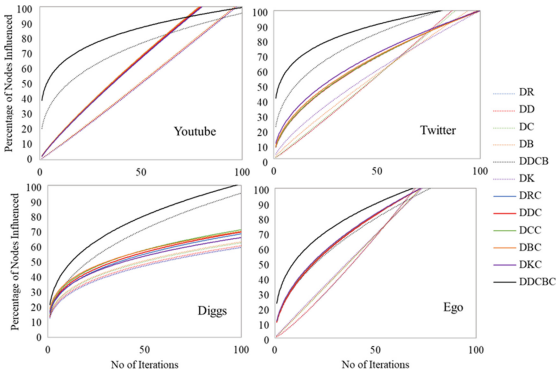

Fig. 6. Percentage of Number of Nodes Influenced in Youtube, Twitter, Diggs and Ego Networks. A Comparison of all methods for 100 iterations.

How Much Advantage Do Community-Level Driver Based Seed Selection Methods Give? From Table 2, we see the percentage of gain that DDCBC has over other seed selection methods in terms of number of nodes influenced after 100 iterations when seed size is 20%. We can see from the table that DDCBC outperforms all methods, but the gain is bigger in terms of global-level driver based methods than the community-level driver based methods. We see this difference in gain mainly because of locally selected and then ranked driver nodes. Also, community creation plays an important role as, the communities are denser than the overall network. From Table 2 we can see that the biggest gain is achieved by DDCBC method over DK method which is 45.89% in FbA network. And the lowest gain is achieved by DDCBC method over DK method in ZKC network. The reason for lowest or lower gain is that ZKC has the highest network density and smallest size. In denser networks, we tend to see the less gain in DDCBC method. Which precisely can mean that, if we locally identify communities, those have denser structures as compared to the overall network. That is why community-driver based methods combined with ranking of DCB works better than the rest of the methods.

5 Conclusion and Future Work

An idea of bringing the methods from control and influence fields together has been proposed in this research. In fact, we played with a research dimension that is at the intersection of both fields and fulfils the objectives of many research questions from both domains. We proposed, implemented and compared a list of new and novel seed selection methods with the traditional seed selection methods from influence domain and driver seed selection methods from influence meets control field. In this work, we introduced new seed selection methods, by utilising driver nodes in communities of the networks. The new methods outperformed the old ones. This opens up an avenue in the already existing research of control

methods in complex networks. Our community-driver based methods show that, we can achieve maximum influence in fewer number of iterations and with a comparatively less seed set size. Also, if we use ranking mechanisms based upon the centrality measures combining degree, betweenness and closeness, the driver nodes selected as seed nodes perform much better in that case as compared to when we rank them on the basis of individual centrality measures.

Work remains to be done in the context of ranking of driver nodes by using different other algorithms for example, Page Rank, Leader Rank, cluster Rank and K-Shell Decomposition. E.g., Page Rank [2], Leader Rank [16], Cluster Rank [3] and K-Shell Decomposition [15]. New methods such as Preferential Matching [27] can be used to identify driver nodes to improve the efficiency of the seed selection process. Another avenue for exploration is the effects of differing influence models, such as the Independent Cascade Model [8].

Acknowledgement. This work was supported in part by the Polish National Science Centre, under Grant no. 2016/21/D/ST6/02408, and in part by the Australian Research Council, Dynamics and Control of Complex Social Networks, under Grant DP190101087.

References

1. Borgs, C., Brautbar, M., Chayes, J., Lucier, B.: Maximizing social influence in nearly optimal time. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 946–957. SIAM (2014)
2. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**(1–7), 107–117 (1998)
3. Chen, D.B., Gao, H., Lü, L., Zhou, T.: Identifying influential nodes in large-scale directed networks: the role of clustering. PLoS ONE **8**(10), e77455 (2013)
4. Chen, Y.C., Zhu, W.Y., Peng, W.C., Lee, W.C., Lee, S.Y.: CIM: community-based influence maximization in social networks. ACM Trans. Intell. Syst. Technol. (TIST) **5**(2), 1–31 (2014)
5. Cheng, S., Shen, H., Huang, J., Zhang, G., Cheng, X.: StaticGreedy: solving the scalability-accuracy dilemma in influence maximization. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 509–518 (2013)
6. Cohen, E., Delling, D., Pajor, T., Werneck, R.F.: Sketch-based influence maximization and computation: scaling up with guarantees. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 629–638 (2014)
7. D'Angelo, G., Severini, L., Velaj, Y.: Influence maximization in the independent cascade model. In: ICTCS, pp. 269–274 (2016)
8. Duan, W., Gu, B., Whinston, A.B.: Informational cascades and software adoption on the internet: an empirical investigation. MIS Q. 23–48 (2009)
9. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. **99**(12), 7821–7826 (2002)
10. Goyal, A., Lu, W., Lakshmanan, L.V.: CELF++ optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th International Conference Companion on World Wide Web, pp. 47–48 (2011)

11. Granovetter, M.: Threshold models of collective behavior. Am. J. Sociol. **83**(6), 1420–1443 (1978)
12. Kazemzadeh, F., Safaei, A.A., Mirzarezaee, M.: Influence maximization in social networks using effective community detection. Phys. A **598**, 127314 (2022)
13. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
14. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 177–187 (2005)
15. Liu, Y., Tang, M., Zhou, T., Do, Y.: Improving the accuracy of the k-shell method by removing redundant links: from a perspective of spreading dynamics. Sci. Rep. **5**(1), 1–11 (2015)
16. Lü, L., Zhang, Y.C., Yeung, C.H., Zhou, T.: Leaders in social networks, the delicious case. PLoS ONE **6**(6), e21202 (2011)
17. Nacher, J.C., Akutsu, T.: Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control. New J. Phys. **14**(7), 073005 (2012)
18. Nguyen, H.T., Thai, M.T., Dinh, T.N.: Stop-and-stare: optimal sampling algorithms for viral marketing in billion-scale networks. In: Proceedings of the 2016 International Conference on Management of Data, pp. 695–710 (2016)
19. Nguyen, H.T., Thai, M.T., Dinh, T.N.: A billion-scale approximation algorithm for maximizing benefit in viral marketing. IEEE/ACM Trans. Netw. **25**(4), 2419–2429 (2017)
20. Nguyen, H., Zheng, R.: On budgeted influence maximization in social networks. IEEE J. Sel. Areas Commun. **31**(6), 1084–1094 (2013)
21. Sadaf, A., Mathieson, L., Musial, K.: An insight into network structure measures and number of driver nodes. In: Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 471–478 (2021)
22. Sadaf, Mathieson, B., Musial: A bridge between influence models and control methods [manuscript submitted for publication]. IEEE Trans. Netw. Sci. Eng. (2022)
23. Sanchis, L.A.: Experimental analysis of heuristic algorithms for the dominating set problem. Algorithmica **33**(1), 3–18 (2002)
24. Sathiyakumari, K., Vijaya, M.S.: Community detection based on Girvan Newman algorithm and link analysis of social media. In: Subramanian, S., Nadarajan, R., Rao, S., Sheen, S. (eds.) CSI 2016. CCIS, vol. 679, pp. 223–234. Springer, Singapore (2016). https://doi.org/10.1007/978-981-10-3274-5_18
25. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: a martingale approach. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1539–1554 (2015)
26. Tang, Y., Xiao, X., Shi, Y.: Influence maximization: Near-optimal time complexity meets practical efficiency. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 75–86 (2014)
27. Zhang, X., Lv, T., Yang, X., Zhang, B.: Structural controllability of complex networks based on preferential matching. PLoS ONE **9**(11), e112039 (2014)
28. Zhu, J., Wang, B., Wu, B., Zhang, W.: Emotional community detection in social network. IEICE Trans. Inf. Syst. **100**(10), 2515–2525 (2017)