Pseudo-Cores: The Terminus of an Intelligent Viral Meme’s Trajectory

Yayati Gupta¹, Debarati Das² and S. R. S. Iyengar¹

¹ Indian Institute of Technology Ropar
yayati.gupta@iitrpr.ac.in, srsiyengar@iitrpr.ac.in
² PES Institute of Technology, Bangalore
debarati.d1994@gmail.com

Abstract. Comprehending the virality of a meme can help us in addressing the problems pertaining to disciplines like epidemiology and digital marketing. Therefore, it is not surprising that memetics remains a highly analyzed research topic ever since the mid 1990s. Some scientists choose to investigate the intrinsic contagiousness of a meme while others study the problem from a network theory perspective. In this paper, we revisit the idea of a core-periphery structure and apply it to understand the trajectory of a viral meme in a social network. We have proposed shell-based hill climbing algorithms to determine the path from a periphery shell (where the meme originates) to the core of the network. Further simulations and analysis on the networks behavioral characteristics helped us unearth specialized shells which we term Pseudo-Cores. These shells emulate the behavior of the core in terms of size of the cascade triggered. In our experiments, we have considered two sets for the target nodes, one being core and the other being any of the pseudo-cores. We compare our algorithms against already existing path finding algorithms and validate the better performance experimentally.

Keywords: Complex Networks, Core-Periphery Structure, Hill Climbing Algorithms, Decentralised Search, K-Shell Decomposition

1 Introduction

Social media captures the essence of an individual’s existence on the internet. In fact, a survey conducted in January 2015 reported that an average user logs in 1.72 hours per day on social platforms, which constitutes about 28 percent of all the online activity of the user³. Such online social transmission has led to the building of very powerful social networks which act as reservoirs of information, holding a huge potential for data analytics. Digital marketing experts and data analysts have harnessed this potential to strategically target some people in the network to spread their information effectively. While social transmission is both frequent and important, it is difficult to understand why some memes are

³ http://www.adweek.com/socialtimes/time-spent-online/613474
more “viral” as compared to others. Some Twitter trends quickly spike in popularity while others languish. What if it is not the meme’s intrinsic contagiousness \[2\] that causes virality but the links in the path of a meme’s trajectory which causes expatiation of information? The power of intelligently made links is a huge contributor to the hidden potential of a social network. The pattern in the connections present in a social network are not only a huge indication of influence points in the network but, also confer different properties to the network itself.

Kitsak et al. have made remarkable observations about influential nodes in his seminal paper \[3\] where they conclude that the “core” nodes in a network are conducive to information amplification. This work motivates us to visualize the network as a meso property suggested by Borgatti and Everett \[4\] known as a “Core-Periphery” structure. Research done on social networks proves that social networks demonstrate scale-free degree distribution \[5\]. Works done by Della et al. \[6\] and Liu et al. \[7\] prove that scale-free networks usually possess a core-periphery structure, transitively we may state that social networks possess core-periphery structure. It would be of immense significance to consider the core as the epicenter of a contagion spreading in the network as suggested by Kitsak et al. Since hitting the core causes great amplification of the contagion, we probe upon the question: “How can we intelligently target some links to quickly reach the core of a network?”. If information usually originates in the periphery of a network, our problem then reduces to an optimum path finding algorithm given the source is the periphery of the network and the core is the destination.

Milgram’s experiment\[8\] had a similar aim to find a shortest path from a source person to the target person. It is not possible to employ a breadth first search approach for such kind of problems since it will lead to flooding of letters in the network. It is also not comprehensible that a person can advertise some product to each of his/her neighbours. Following a DFS is also quixotic since the user may follow a totally wrong path. It is also seen that while BFS results in a higher space complexity, DFS results in a higher time complexity \[9\]. It was observed that even though people did not possess an overview of the entire network, they were still able to trace the average 6-hop path between two individuals. This spawned the idea of a decentralised search approach, the special case of which is the \[10\] “Myopic Search” approach. This greedy heuristic aimed at providing a path from a source to a destination exploring only one new node per iteration which is nearest to the target. We realised the need for a similar intelligent decentralised search based method to direct the meme in an optimal direction. Therefore, we propose a hill climbing technique by virtue of which a user needs to focus only on one neighbour. This neighbour gives him/her more benefit instead of distributing his/her efforts among all the neighbours.

We evaluate various properties of the shells provided by the k-shell decomposition algorithm. Employing the properties of these shells might help us take an intelligent walk to easily reach our destination. We also discover shells which mimic the cores and help a meme go viral. We call these shells as \[“Pseudo-\]
Cores”. These revelations of our experiments have tremendous impact on the fields of information propagation as well as epidemiology. Not only can intelligent pathways be formulated to propagate information, but also if such pathways could be simulated, preventive checkpoints could be placed appropriately to halt infection spread.

2 Preliminaries

2.1 Core Periphery Structure

Core-Periphery structure is a meso-scale property found in social networks. In general, high status individuals are easier to target in a population essentially because they have an access to higher privileges compared to the common crowd. They are connected over social boundaries while lower status people are mostly clustered and display local links. This distinction arises because of homophily (“birds of a feather flock together”). The high status people tend to form communities like the low status people. But, these two groups do not occupy interchangeable positions in a social network. High status people intersect in a densely connected core, while the low status people reside in the periphery of the network. Periphery nodes separated by a large distance tend to pass through the core to connect to each other. According to Rossa et al. [6], scale-free networks inherently possess core-periphery structure.

2.2 K-Shell Decomposition

This algorithm was proposed by Kitsak et al. [3] in order to determine the core in a network. The core of a network can be defined as a set of densely connected nodes which collectively possess a high closeness centrality. The algorithm works by recursively pruning the lower degree nodes in a network. First the nodes having degree $d(u) \leq 1$ are pruned recursively till there are no degree 1 nodes left in the network. Similarly, the process is repeated for the nodes $u$ having degree $d(k)$ where $k=2,3,\ldots,n-1$ till the graph becomes empty. At every step, the pruned nodes are kept in a basket. The baskets formed earlier represent the least core nodes and vice versa. The nodes in one basket form one shell of the graph and a higher shell number represents higher coreness. The intricacy of the algorithm lies in the fact that the nodes having high degrees lying at the periphery are pruned earliest because of recursive pruning. Therefore, it is not necessary that a person having high degree should also be a core node.

3 Motivation Inciting the Algorithm

A meme is like a virus which lives in secrecy until it grows so much in multitude that it sweeps the network by the sheer weight of its population. It supplements with other hosts of the virus and uses their resources to increase its popularity until it reaches the right atmosphere where it grows exponentially. The cause for
meme virality has been attracting a lot of attention and research since the mid 1990s. Many theories have been proposed for understanding this phenomenon, some of which focus on the content of the meme, while others on its spreading pattern. We focus on the network topology aspect of meme propagation and observe, if following a particular set of links in the network might bring us closer towards infecting the influentials in a network. This transitively leads to the information being propagated across the entire network. We term the potential of a node to infect the entire network and cause amplification of information as the node’s ‘spreading power’. We use k-shell decomposition algorithm to reduce a social network into multiple shells. The outermost shells denote the least degree of “coreness” and the innermost shell is termed as the core of the network. The core has the highest spreading power. We visualise the entire network as a core periphery structure and observe how network properties vary across shells. From this perspective, we have also performed experiments to illustrate our algorithms. The datasets used for these experiments are given in Appendix.

3.1 Experiments undertaken to investigate the correlation of network properties with the shell number

Distribution of nodes across shells: The plot in figure 1 indicates that the number of nodes do not necessarily follow an ideal rectangular hyperbola curve as intuitively expected (with maximum nodes in the periphery and minimum in the core). There is no hierarchy shown in the distribution of the population of nodes across shells. The plot shows increase in the population of nodes in the intermediary shells as well. As the core is the most densely connected and centrally reachable portion of the network, it is the shell with the least diameter but it does not necessarily have the least number of nodes.

![Distribution of nodes across shells](image)

Fig. 1. Distribution of nodes across shells for various networks
**Density of shells:** Buzznet, Slashdot, Livemocha, Flickr and Google Plus are some social networks which show an uniformly exponentially increasing curve of density distribution. A similar curve is observed in the case of collaboration network DBLP. When the core shell is encountered in these networks, there is a sudden but powerful spike in density. This is probably one of the contributory factors to the spreading power of the core. However, in the case of social networks like Facebook, the curve does not accelerate only on reaching the core, there are quite a few spikes in between and the curve does not remain monotonically increasing. However, even in this case, core is observed to possess the maximum density. As a result we can safely assume that the high density of the core is the reason for the high spreading power of the core. The plots are shown in figure 2.

(a) Density of shells in buzznet, slashdot and livemocha  
(b) Density of shells in DBLP  
(c) Density of shells in Flickr, Google Plus and Facebook

**Fig. 2.** Shellwise Density distribution in Real world networks

**Cascading power of Different shells in a network** The plots in figure 3 represent the meme cascade size produced if the cascade starts from some of the nodes of a particular shell. Independent cascade model is used for simulation with equal probability of infection transmission across every edge. One ideally expects the cascade to accelerate when core shell is encountered, but it is observed that the acceleration point is reached much before meme’s reaching the core.
This astonishing fact led us to investigate the existence of a “pseudo- core shell” or a shell which provides something akin to an escape velocity for the meme to become viral. Once this shell has been visited in the meme trajectory, the meme goes viral and there is no longer any need for it to target some core node. This hypothesis has many large scale implications. For example- If I were a political analyst and I were trying to find which person to infect in a political network, I would no longer have to infect the most influential politicians or relatively insulated core nodes. Infecting someone relatively less influential (if I could find that this person lies in a pseudo core shell) would cause the same effect.

We compare a set of path finding algorithms in the next section and further observe if changing the destination to pseudo core has any relative impact on the time taken for the path.

4 Algorithms

We describe algorithms in this section to find a path from the periphery of a network to the destination, which we initially define as core. Later we apply the same algorithm with pseudo-cores as the destination and report the improved results. We describe two already implemented algorithms in Table 1.

Below, we propose two hill climbing algorithms based on the shell number of the nodes. These algorithms utilize the concept of k-shell decomposition to reduce the network into shells. The entire system can be visualized as a circular maze made up of concentric circles (shells) where the goal of the algorithm is to intelligently move from the outermost shell to the innermost shell. There are inter-shell edges that help a user in taking such a walk across shells, while the intra-shell edges helps the user to traverse a shell.

Algorithm 1- Shell based Hill Climbing Approach (SH): Here $G(V,E)$ represents the graph where $V(G)$ is the set of vertices and $E$ is the set of edges.
Random walk algorithm | Degree based hill climbing
---|---
This algorithm involves a node inspecting its neighbours at every step and selecting one of them randomly. If the chosen neighbour is a core node, the algorithm terminates, else the selection of the random neighbours continues. Random walk algorithm (without repetition of nodes) has a time complexity of $O(n)$. | This algorithm uses a hill climbing approach based on the degree of the nodes in the network. At every step, a node looks at its neighbours and chooses the unexplored node having the highest degree. If the chosen node is a core node, the algorithm terminates, else the process continues. As hill climbing algorithms have a complexity of $O(n)$ where $n$ is the number of nodes and finding degree of all nodes takes $O(m)$ time, degree based hill Climbing has a time complexity of $\max[O(n), O(m)] \sim O(n)$ in sparse graphs.

Table 1. Existing algorithms

$shell(u)$ represents the shell number of a node $u$ as calculated by the k-shell decomposition algorithm. $start$ is the periphery node from where the meme starts spreading. The proposed SH approach has a complexity of $\max[O(m + n), O(n)] \sim O(n)$ in sparse graphs.

Algorithm 1 Shell Based Hill Climbing (SH)

```
procedure FindNumSteps
    Input: Graph $G(V, E)$, Starting node $start$
    Output: Number of steps taken by the algorithm to terminate
    Apply k-shell decomposition and calculate $shell(u) \ \forall u \in V(G)$
    $visited[u] \leftarrow \text{false} \ \forall u \in V(G)$
    $numsteps \leftarrow 0$
    $current \leftarrow start$
    $visited[current] \leftarrow \text{true}$
    while $current$ is not a core node do
        $v_1 \leftarrow \text{argmax}_{u \in V(G), visited[u] = \text{false}} shell(u)$
        if $shell(v_1) \leq shell(current)$ then
            $v_2 \leftarrow \text{random neighbour u of current such that visited[u] = \text{false}}$
            $current \leftarrow v_2$
        else
            $current \leftarrow v_1$
        $numsteps \leftarrow numsteps + 1$
    return $numsteps$
```

Algorithm 2 - Intershell Hill Climbing with Intrashell Degree Based Approach (SA): Algorithm 2 is a modification of algorithm 1 and utilises the idea that a node with very high degree will cover most of the shell. If this node is chosen, it would greatly reduce the number of steps required to traverse a shell.
The proposed SA approach has a complexity of $\max[O(m + n), O(n), O(m)] \sim O(n)$ in sparse graphs.

\begin{algorithm}
\caption{Improved Shell Based Hill Climbing (SA)}
\begin{algorithmic}[1]
\Procedure{FindNumsteps}{ } 
\Input{Graph $G(V, E)$, Starting node $start$} 
\Output{Number of steps taken by the algorithm to terminate} 
\State Apply k-shell decomposition and calculate $shell(u) \ \forall u \in V(G)$ 
\State $visited[u] \leftarrow \text{false} \ \forall u \in V(G)$ 
\State $numsteps \leftarrow 0$ 
\State $current \leftarrow \text{start}$ 
\State $visited[current] \leftarrow \text{true}$ 
\While{$current$ is not a core node} 
\State $v_1 \leftarrow \text{argmax}_{u \in V(G) \land visited[u]='false'} shell(u)$ 
\If{$shell(v_1) \leq shell(current)$} 
\State $v_2 \leftarrow \text{argmax}_{u \in V(G) \land visited[u]='false'} degree(u)$ 
\State $current \leftarrow v_2$ 
\Else 
\State $current \leftarrow v_1$ 
\EndIf 
\State $numsteps \leftarrow numsteps + 1$ 
\EndWhile 
\State return $numsteps$
\EndProcedure
\end{algorithmic}
\end{algorithm}

(a) Algorithm1- Shell Based Hill Climbing (SH)  
(b) Algorithm2- Modified Shell Based Hill Climbing (SA)

Fig. 4. Proposed Algorithms: The path denoted in the pink edges is the path chosen by the corresponding algorithm to move towards the core
5 Experimental Results

To evaluate the performance of the algorithms mentioned in the above section, we select periphery nodes from shell 1 (periphery) in a network and for each of these nodes, we find the number of steps taken to reach the core. We term each run from a periphery node as an instance of the problem. Therefore, we can say the number of instances is equal to the number of nodes. It is observed that more than 80% of the walks conclude in a maximum of 15 steps in most of the datasets. We have also disregarded the trivial case where source nodes are directly connected to the core as the path length in these cases is 1.

Let $R$ be a random variable whose value ranges from 2 to $k$. $R$ depicts the number of steps taken by the algorithm to terminate. Let $P(R = k)$ be the probability of value of $R$ being $k$ where $k$ ranges from 2 to 15. We plot the cumulative probability distribution function of $R$. X axis indicates all possible values of $R$ while Y axis shows the probability of $R \leq k$.

The plots given below validate that the proposed algorithms cover most of the instances in very less number of steps as compared to the existing path finding algorithms. The highest line in the curve represents the most efficient algorithm. In the case of Facebook network, the proposed algorithms cover 80% of the instances in less than 100 steps. Degree based hill climbing requires around 200 steps to cover 80% of the instances. In the case of Google Plus, all the three hill climbing based approaches cover 90% of the instances in less than 3 steps. The results for the rest of the networks are included in Appendix. In all the cases, the algorithms proposed reach their peak at the earliest proving that they are more optimal with respect to time taken to reach destination. The random walk algorithm clearly performs the worst.

![Comparison of algorithms for Facebook](image)

(a) Random Walk  (b) Shell Based Hill Climbing Algorithms

Next, we modify the destination to be the pseudo core shells and observe the cumulative frequency distribution of $R$ as given in figure 7. In this case also, our proposed algorithms perform better than the other algorithms. Interestingly, the performance of even the random walk algorithm increases drastically when the target is changed to pseudo-cores. This indicates that the virality which seems frequent and random in our social as well as biological networks may be because
of the presence of pseudo-cores in the network. It is fairly intuitive that it is
difficult to target an insulated and well connected core node but it would be
relatively simple to hit the pseudo-core and this could be one of the possible
explanations for meme virality in a network. The results for these simulations
are shown in figure 7.

6 Related Work

Genes display a very well known phenomenon informally stated as “Survival of
the Fittest” [12]. They replicate and mutate to propagate evolution. Dawkins
suggested that a similar system applies to information and ideas, which he termed as “memes”. Like genes, memes also undergo cultural evolution as explained by Heylighen et al. [13]. Analogies have been observed between a viral meme in a social network and an epidemic in a biological contact network [15][16]. Memetics has many applications in the fields of digital marketing as suggested by Leskovec et al. [8]. The information potential of a social network has been harnessed by Culotta et al. [17] who have used it to predict epidemics in a population. The information flowing through these sites have been used to predict the results of election [18] and stock markets [19] as well as for predicting the crime prevalence [20].

Many approaches have been taken to understand the causes of meme virality. Berger & Milkman [2] employed the content of a meme to predict its virality. Weng et al. did a seminal work in this direction and observed the similarity between a simple contagion and a viral meme [11]. The existence of communities and core-periphery structure [4] are two major discoveries with respect to complex network structure. We have applied our knowledge of complex network structures to understand meme spread. Standing on the shoulders of all the work discussed above, the question we probed on was: “Can we intelligently alter the path of a meme flowing through a network to make it go viral?”. Kleinberg et al. [10] attempted targeting the optimal selection of seed nodes [1], but as yet no work has been done to identify the optimum destination nodes in information propagation, which is what we are attempting empirically. Though work has been done on path finding algorithms in a social network, we are unaware of any approaches which have combined the idea of visualizing a network as a core-periphery structure and utilizing this property to devise a path finding algorithm over the network. Our work is the first of its kind to the best of our knowledge.

7 Conclusion

The paper attempts to analyse the impact of the characteristics of shells in a core-periphery network on the virulence of a meme. We empirically observed that the innermost core shell has the greatest tendency to trigger a global cascade in the network, thereby increasing the necessity to infect the core quickly in order to cause virality. We have proposed two shell based hill climbing approaches that help a meme to pave an intelligent path to the core, when it originates in the periphery. One of the most important contributions of the paper is the unveiling of the concept of “Pseudo Core” shells that have the same cascading impact on the network as a core shell. Intelligently hitting the pseudo core shell achieves the same virality as that achieved by core shell. As a result the path taken during the trajectory of a viral meme can be reduced. These revelations introduced by our experiments have huge impact across several disciplines.
8 Future Work

While performing experiments to analyse network property effects on shells, we defined a shell parameter which we deem “Leakage Power”. Leakage power denotes a shells potential to take quick and long jumps to the higher numbered shells. We plotted leakage power against shell number and observed that the leakage power was not necessarily the highest in the case of core. There were indications of high leakage power in intermediary shells as well. This is shown in figure 8. This suggested to us possible ideas of “teleportation shells” and “barricade shells”. The shells having higher leakage powers act as the teleportation shells and can trigger a meme to take longer jumps on its path to the core. On the other hand, the shells having low leakage powers may tend to block a meme inside it, hence suggesting why some memes are non viral. Based on these observations, the algorithms may be altered to provide better results and most importantly answer the bigger question which is “Why do some memes selectively go viral ?”. Another interesting research idea is to come up with an approximation algorithm to determine the coreness of a node based on local information. Approximating the coreness/shell number locally may improve the time complexity of core finding algorithms.

References

1. D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003, pp. 137–146.
2. J. Berger and K. L. Milkman, “What makes online content viral?” Journal of marketing research, vol. 49, no. 2, pp. 192–205, 2012.
3. M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, “Identification of influential spreaders in complex networks,” *Nature Physics*, vol. 6, no. 11, pp. 888–893, 2010.
4. S. P. Borgatti and M. G. Everett, “Models of core/periphery structures,” *Social Networks*, vol. 21, no. 4, pp. 375–395, 2000.
5. A.-L. Barabási *et al.*, “Scale-free networks: a decade and beyond,” *science*, vol. 325, no. 5939, p. 412, 2009.
6. F. Della Rossa, F. Dercole, and C. Piccardi, “Profiling core-periphery network structure by random walkers,” *Scientific reports*, vol. 3, 2013.
7. R. Zafarani and H. Liu, “Social computing data repository at ASU,” 2009. [Online]. Available: [http://socialcomputing.asu.edu](http://socialcomputing.asu.edu)
8. S. Milgram, “The small world problem,” *Psychology today*, vol. 2, no. 1, pp. 60–67, 1967.
9. R. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972.
10. J. Kleinberg, “Complex networks and decentralized search algorithms,” 2006.
11. L. Weng, F. Menczer, and Y.-Y. Ahn, “Virality prediction and community structure in social networks,” *Scientific reports*, vol. 3, 2013.
12. M. Radman, F. Taddei, and I. Matic, “Evolution-driving genes,” *Research in Microbiology*, vol. 151, no. 2, pp. 91–95, 2000.
13. R. Dawkins, “Viruses of the mind,” *Dennett and his critics: Demystifying mind*, pp. 13–27, 1993.
14. F. Heylighen, “What makes a meme successful? selection criteria for cultural evolution,” 1998.
15. D. J. Daley and D. G. Kendall, “Epidemics and rumours,” 1964.
16. A. Saxena, S. Iyengar, and Y. Gupta, “Understanding spreading patterns on social networks based on network topology,” *arXiv preprint arXiv:1505.00457*, 2015.
17. A. Culotta, “Towards detecting influenza epidemics by analyzing twitter messages,” in *Proceedings of the first workshop on social media analytics*. ACM, 2010, pp. 115–122.
18. E. T. K. Sang and J. Bos, “Predicting the 2011 dutch senate election results with twitter,” in *Proceedings of the Workshop on Semantic Analysis in Social Media*. Association for Computational Linguistics, 2012, pp. 53–60.
19. X. Zhang, H. Fuehres, and P. A. Gloor, “Predicting stock market indicators through twitter i hope it is not as bad as i fear,” *Procedia-Social and Behavioral Sciences*, vol. 26, pp. 55–62, 2011.
20. M. S. Gerber, “Predicting crime using twitter and kernel density estimation,” *Decision Support Systems*, vol. 61, pp. 115–125, 2014.
21. J. Leskovec and J. J. McAuley, “Learning to discover social circles in ego networks,” in *Advances in neural information processing systems*, 2012, pp. 539–547.
22. J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, “Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters,” *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.
23. J. Yang and J. Leskovec, “Defining and evaluating network communities based on ground-truth,” *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.
A APPENDIX

A.1 Datasets Overview

Table 2 gives a short description of the all the datasets used in this paper.

| Dataset     | Description                                                                                                                                 |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Facebook    | Facebook is the most popular Social Networking Site today. This dataset consists of anonymized friendship relations from Facebook. [21]. The network contains 4,039 nodes and 88,234 edges. |
| Google plus | Google plus is a social layer for Google Services. [21]. The network contains 107,614 nodes and 13,673,453 edges.                              |
| Slashdot    | Slashdot is a website where the users can submit and evaluate the news stories on science and technology. It is famous for its specific user community. [22]. This dataset contains 82,168 nodes and 948,464 edges. |
| Flicker     | This is an image and video hosting site. It is mainly used for sharing and embedding personal photographs. [7]. This dataset has 80513 nodes and 5899882 edges. |
| Livemocha   | Livemocha is the world’s largest online language learning community. [7]. This dataset has 104438 nodes and 2196188 edges.                     |
| DBLP        | The DBLP computer science bibliography is a collaboration network. It provides a detailed list of research papers in computer science. [23]. The network contains 317,080 nodes and 1,049,866 edges. |
| Buzznet     | Buzznet is social media network used for sharing photos, journals, and videos. It has 101168 nodes and 4284534 edges.                          |

Table 2. Datasets used for experiments

A.2 Leakage Power

This definition formally defines the leakage power introduces in the future section of the paper. For defining the leakage power, first we define a ”Teleportation Edge”:

**Teleportation Edge**: An edge $E_{ij}$ in the network is called a teleportation edge if $shell(j) > shell(i)$.

The leakage power of a shell $S$ is proportional to the number and the average height of the ladders present in this shell. Let the vertex set of a shell $s$ be represented as $V(S)$ and edge set as $E(S)$. Let the leakage power of this shell represented by $\theta_S$.

Then, $\theta_S \propto |E_{ij}|$ where $i \in V(S)$.

Let the height of a ladder $E_{ij}$ be defined by $H_{ij} = shell(j) - shell(i)$

Then, $\theta_S \propto \sum_{E_{ij} \in E(S)} \frac{H_{ij}}{m \times (n-m)}$. where $m = |V(S)|$ and $n = |V(G)|$
So, overall
\[ \theta_S = \kappa |E_{kl}| \sum_{e_{ij} \in E(S)} H_{ij} \frac{m \times (n-m)}{m \times (n-m)}. \]
where \( k \in V(S), m = |V(S)|, n = |V(G)|, \)
and \( E_{kl} \) is a ladder.

\( \kappa \) is a parameter whose value depends on the network and needs to be suitably found out.

### A.3 Results of the proposed algorithms

The figures 9, 10 and 11 show the results of the algorithms discussed in the paper for three networks for the case when the destination is a core shell.

![Image](image1.png)

(a) Random Walk

![Image](image2.png)

(b) Shell Based Hill Climbing Algorithms

**Fig. 9.** Comparison of algorithms for Slashdot
Fig. 10. Comparison of algorithms for DBLP

(a) Random Walk

(b) Shell Based Hill Climbing Algorithms
Fig. 11. Comparison of algorithms for Livemocha

(a) Random Walk

(b) Shell Based Hill Climbing Algorithms