ShatterPlots: Fast Tools for Mining Large Graphs

Ana Paula Appel
Computer Science Department
USP at São Carlos - ICMC - Brazil
anaappel@icmc.usp.br

Deepayan Chakrabarti
Yahoo! Research
Sunnyvale, CA
deepay@yahoo-inc.com

Christos Faloutsos
School of Computer Science
Carnegie Mellon University
christos@cs.cmu.edu

Ravi Kumar
Yahoo! Research
Sunnyvale, CA
ravikuma@yahoo-inc.com

Jure Leskovec
Computer Science Department
Cornell University
jure@cs.cornell.edu

Andrew Tomkins
Yahoo! Research
Sunnyvale, CA
atomkins@yahoo-inc.com

Abstract

Graphs appear in several settings, like social networks, recommendation systems, computer communication networks, gene/protein biological networks, among others. A deep, recurring question is "*What do real graphs look like?*" That is, how can we separate real ones from synthetic or real graphs with masked portions? The main contribution of this paper is **ShatterPlots**, a simple and powerful algorithm to extract patterns from real graphs that help us spot fake/masked graphs.

The idea is to shatter a graph, by deleting edges, force it to reach a critical ("Shattering") point, and study the properties at that point.

One of the most striking patterns is the "*30-per-cent*": at the Shattering point, all real and synthetic graphs have about 30% more nodes than edges. One of our most discriminative patterns is the "*NodeShatteringRatio* ", which can almost perfectly separate the real graphs from the synthetic ones of our extensive collection.

Additional contributions of this paper are (a) the careful, scalable design of the algorithm, which requires only $O(E)$ time, (b) extensive experiments in a large collection of graphs (19 in total), with up to hundreds of thousands of nodes and million edges, and (c) a wealth of observations and patterns, which show how to distinguish synthetic or masked graphs from real ones.

1 Introduction

Graphs appear in numerous settings, like social networks, scientific publication network, conferences vs. authors, among others. The aim of this study is to find patterns to help us spot fake and "masked" graphs. (By "*masked*" we mean a graph that is a non-random sample of a real graph - for example, a real graph after one has deleted all the nodes with degree ≤ 100). It proposes to extract the characteristics from large graphs with the novel tool of ShatterPlots. Moreover, the method is expected to be scalable, so that it can handle graphs that span MegaBytes, GigaBytes or more.

The main idea behind ShatterPlots resembles high-energy physics, where particles are smashed, and experts study the results of the collisions to reach conclusions. Here, the proposal is to shatter the given graph, that is, to drive it to the "**Shattering point**", by deleting edges at random, and observing its behavior. The first research challenge is how to interpret the results of the Shattering, and the second is scalability and speed.

The answers to the above challenges are exactly the contributions of this work. For the first, the study shows that random edge deletion always leads to a high spike of the diameter, exactly at the critical point called "Shattering point".

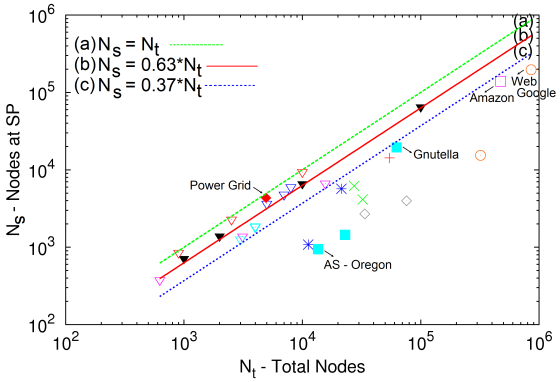


Figure 1: Our *NodeShatteringRatio* pattern allows an impressive distinction between fake/masked graphs (triangles, Amazon, Web Google) and real graphs (rest).

At the Shattering point, a list of surprising observations for several real graphs is given. The most surprising is the

"30 per cent" pattern, which states that under a random edge deletion, real graphs have 30% more nodes than edges when they reach their Shattering point, *regardless* of the original graph. Another interesting observation is that at the Shattering point, the count of the remaining edges is $1/\lambda_1$ of the original edge count, where λ_1 is the first eigenvalue of the original graph. It is fascinating that $1/\lambda_1$ is the epidemic threshold of graph [12].

The most striking pattern is the *NodeShatteringRatio* , illustrated in Figure 1. This pattern allows perfectly separating the real graphs from fake/masked ones, at least for the graphs of our collection. Specifically, the fraction N_s/N_t of the remaining nodes at the Shattering point is much much lower for most real graphs, while it is about 0.7 for the masked ones (and for the Erdős-Rényi graphs). (N_s is the number of nodes at the Shattering point and N_t is the total number of nodes of the original graph.)

Finally, for scalability, a fast and adaptive algorithm that can quickly discover the Shattering point is proposed. Its performance is linear on the number of edges E, as shown empirically.

The paper is organized as follows: Section 2 surveys the related techniques; Section 3 proposes the data model and the formal problem specification and further presents the algorithms; Section 4 evaluates the algorithms with real data; Section 5 and 6 present the patterns found and proofs and outliers spotted. The scalability is presented in Section 7 and Section 8 presents the conclusions.

2 Related Work

There is a significant body on research related to our problem, which is categorized into the following groups: graph algorithms, graph patterns, epidemiology, phase transitions, and outliers detection.

Graph Algorithms: Intuitively we expect the graph to shatter at the point where natural communities or clusters break apart. Popular methods for partitioning graphs include the METIS algorithm [25], spectral partitioning techniques [24], flow-based methods [21] information-theoretical methods [15], and methods based on the "betweeness" of edges [23], among others. Note that our work is orthogonal to this, given that fast and scalable techniques are used to examine the structure of the graph. Probably the most related work is the k-cores [8] decomposition, which recursively "peels" the graph. A recent extension for bipartite graphs uses the KNC plots [30]. This approach would be complementary to ours, since the authors examine different aspects of the graph.

Graph patterns: Several old and recent patterns have been discovered for large, real graphs.

The first is the *skewed degree distribution* phenomenon, with power law tails, for the Internet [20], the Web [27, 9], citation graphs [41], online social networks and many others. Deviations from the power-law pattern have also been noticed [39], but the distribution is still very skewed.

The second is the *Small diameter:* This is the the "small-world" phenomenon, or 'six degrees of separation' [48] The diameter of a graph is d if every pair of nodes can be connected by a path of length of at most d. Following the computer network literature, the *effective diameter* [46] is used: The minimum number of hops in which some fraction (or quantile q, typically $q = 90\%$) of all connected pairs of nodes can reach each other. The effective diameter has been found to be small and decreasing over time for large real-world graphs, like Internet, Web, and social networks [3, 36, 32].

Phase transitions: The point where the graph shatters is ultimately a point of phase transition, i.e., a point where the connectivity structure abruptly changes. The Erdős-Rényi graphs exhibit phase transitions [18] in the size of the largest connected component. Several researchers argue that real systems are "at critical points" [6, 45], like avalanches, forests (with forest fires), mechanical tension causing earthquakes, among others. If this also holds for real networks, then they should be ready to "shatter", after few edges deletion. The work presented in [14] makes a relevant study about robustness of network topologies in regular graphs. Phase transition is also known as bond and site percolation threshold. An example of its application is presented in [29].

Epidemiology: Most of the previous researches on the flow of information and influence through the networks has been done in the context of epidemiology and the spread of diseases over the network [5, 12].

The work on spread of diseases in networks and immunization mostly focuses on determining the value of the *epidemic threshold* [5], a critical value of the virus transmission probability above which the virus creates an epidemic.

The epidemiology community has developed the so-called *SIR* and *SIS* models [5] of infection. The *SIS* model (*Susceptible – Infective – Susceptible*) is suitable for the common flu, where nodes may be infected, healed (and susceptible), and infected again.

A recent study has showed that the epidemic threshold of a graph is $1/\lambda_1$, that is, the inverse of its highest eigenvalue [12]. More details will be provided further, as well as their connection with the bond percolation threshold.

Outliers detection in graphs: Finally, we focus on outlier detection, as the connectivity structure revealed by the ShatterPlots. Autopart [11] finds outlier edges in a general graph, however, we need to detect outliers *nodes*. Noble and Cook [37] studied anomaly detection in general graphs with labeled nodes, however, their goal was to identify abnormal substructures in the graph, not abnormal *nodes*. Aggarwal and Yu [2] proposed algorithms to find outliers in high-dimensional spaces, but their applicability to graphs is unclear: the nodes in a graph lie in a vector space formed by the

graph nodes themselves, so the vector space and the points on it are related. As mentioned further, it is possible to observe very different patterns of shattered graphs when compared to simple models, which allows detecting masked/fake non-realistic graphs.

3 Proposed Method

We start with the problem definition and the motivating questions. Then our design decisions are discribed, and our algorithm is finally given.

3.1 Problem Definition. The aim is to find patterns at the Shattering point, which is a clear spike in the diameter after some edges deletion in real graphs, like social networks, citation and web graphs and recommendation systems (users-to-products bipartite networks). The focus of this paper is on the analyses of whether fake/masked graphs have a different behavior than the real graphs at the Shattering point. What can we say about real graphs at the Shattering Point? Can we find interesting patterns in real graph at this point? Can we use these patterns to spot fake/masked graphs?

The problem is defined as follows:

PROBLEM 1. Given *a large, sparse graph check whether it is masked or synthetic graph.*

In fact, there are two types of questions that should be verified for all graphs. The first are "philosophical" questions, whose answers will settle some conjectures. The second set consists of "exploratory" questions, which refer to what properties are expected to be seen, at the Shattering point of a graph (assuming that it does have a Shattering point).

3.1.1 "Philosophical" Questions

PHQ 1. *Do real graphs have a Shattering point?*

Real networks are very resilient [4] at random node deletions while some others, like Erdős-Rényi are not. One would expect so, if we had random edge deletion (*RED*). However are there exceptions in real graphs? Is it possible to have a real graph, whose diameter increases continuously, without an abrupt shattering under *RED*?

PHQ 2. *Are real-life graphs just a bit above the Shattering point?*

One would expect so. For example, Bak [6] proposed the theory of SOC (Self-Organized Critically), arguing that several phenomena are just at their critical point, like avalanches, finances of interrelated companies and tectonic plaques. Several graph generators also focus on 'optimized tolerance' [10, 19]. Thus one might expect real graphs to be connected, but barely so, and thus would be just above Shattering. A communication network that is a way above a Shattering point, would be wasting resources, one might argue.

3.1.2 Exploratory Questions.

Jumping ahead, it turns out that all the real and synthetic graphs that were tried in this study, do have a sharp Shattering point, bringing about a number of questions:

EXQ 1. *What is the Edge shattering ratio E_s/E_t (*i.e.*, the fraction of edges at the Shattering point)? Does it depend on the graph size at all?,*

where E_s is the number of edges and N_s is the number of nodes, both at the Shattering point. E_t is the total number of edges of the original graph and N_t is the total number of nodes in the original graph. The symbols are defined in table 1.

EXQ 2. *What about the Node shattering ratio N_s/N_t (*i.e.*, the fraction of nodes at the Shattering point)?*

EXQ 3. *Do synthetic graphs have the same behavior at the Shattering point? or do they follow different laws?*

EXQ 4. *What can we say about the node-to-edge ratio of a graph at the Shattering point? And about the giant connected component at the Shattering point?*

3.2 Design decisions. *T*hinning methods: Several thinning methods were tried, like Random Edge Deletion(*RED*), and several versions of "Hostile" edge deletion. The most striking patterns were found in the former, thus we shall exclusively focus on *RED* here.

*C*hoice of shattering criterion: The shattering criterion should ideally have a sharp transition. We considered several shattering criteria:

- Size (number of nodes) of the largest weakly connected component
- Effective diameter (number of hops at which 90% of all reachable pairs do reach each other)
- Total number of reachable pairs of nodes

The graph was expected to shatter at all of the above criteria, *i.e.*, there will be a Shattering point in the edge deletion process, where the connectivity of the graph will be seriously disrupted: e.g., the graph becomes disconnected, the size of the largest component drops, the diameter spikes, and the number of reachable pairs of nodes drops. The results of the shattering of our 19 network datasets will be examined in more detail in the following section.

3.3 Algorithm description Next, the algorithm for creating ShatterPlots is presented. However, instead of starting with the full graph and deleting edges at random, the algorithm starts with an empty graph and inserts edges at random. Algorithm 3.1 shows the details.

The idea is to shuffle the edges file of a graph G and builds the temporary graph H adding some numbers of edges

ALGORITHM 3.1. Adaptive ShatterPlot

Input: Input graph $G(N, E)$
Output: Point of shattering (and stats about it)
  Shuffle the $|E|$
  Temporary $H(N,\emptyset)$, on N nodes
  $\epsilon = 0.005$ or $\epsilon = 1/\lambda_1$
  $t = 0$
  $Step(t) = \epsilon * |E|$
  while $H\ != G$ **do**
    Insert $Step(t)$ edges in H at random
    $t = t + 1$
    Measure the structural properties of H (diameter, connected components, first eigenvalue, etc.)
    D_t = effective diameter of H
    if $t > 1$ **then**
      if $D_t - D_{t-1} \geq 1$ **then**
        $Step(t) = Step(t-1)/2$
      else if $D_t - D_{t-1} \leq -1$ **then**
        $Step(t) = 2 * Step(t-1)$
      end if
    else if $\epsilon = 1/\lambda_1$ **then**
      $Step(t) = 0.005 * |E|$
    end if
  end while

Figure 2: ShatterPlot algorithm

($Step(t)$) at random. Both have the same nodes (N), and will be exactly the same at the end of the algorithm. After each insertion we measure the structural properties of the graph, like diameter, number of reachable pairs of nodes, number of triangles, first eigenvalue of the graph adjacency matrix ond size of the largest connected component. The process is repeated until the graph has been full, *i.e.*, has contained all edges from the edges file.

Ideally graph properties should be re-computed after the insertion of each and every edge. Such an approach would be slow, therefore a batch of edges must be inserted at a time. The question is what is the appropriate size of such a batch, so that the Shattering point will not be overshot and missed? Our answer is an adaptive method: we start with a small batch size, and if there is no major difference in the graph structure (say, the diameter), the batch size is increased. Conversely, it is decreased, if a spike seems to be reached. The same process could be applied the other way round, that is, instead of inserting edges, we could start with a full graph and delete edges at random on it. Empirically, the algorithm is very fast, and usually needs about 250 steps to locate the Shattering point.

Scalability: next, we show that the Adaptive ShatterPlot Algorithm scales well on the number of total edges E_t, demonstrating that the Adaptive ShatterPlot is capable of handling large graphs. It scales even better, up to 8 times faster, using the *Eigenvalue* pattern presented in Section 6.

First, edge insertion is assumed to be a constant time operation. This is true for most graph implementations. In some implementations it can be logarithmic/linear in the average degree of the graph, but as real graphs are sparse this is practically constant. Second, due to the Approximate Neighborhood Function algorithm [38] (ANF), it is responsible from calculating the effective diameter of the graph in linear time $O(E)$ on the number of edges E in the graph.

DEFINITION 1. *The effective diameter is the minimum number of hops in which 90% of all connected pairs of nodes can reach each other.*

Also, the effective diameter is a more robust measurement of the pairwise distances between nodes of a graph.

However, this does not solve the problem immediately: if we use a naive implementation of the ShatterPlot algorithm and at every step add a constant number of edges, then the full algorithm would scale quadratically with the number of edges $O(E^2)$ ($O(E)$ for the number of ShatterPlot iterations, and a factor of $O(E)$ for running ANF at each step). Due to the adaptive nature of our algorithm, which exponentially adjusts the number of edges it adds from the graph, we only need a roughly *constant* number of iterations, which makes our algorithm scale well to the number of edges.

There are two versions of ShatterPlot algorithm. The first is called *Proportional ShatterPlots*, in which the initial value ϵ is 0.005 in $Step(0)$. The other version is called *Eigenvalue ShatterPlots*, given that $1/\lambda_1$ is used as the initial value for ϵ at $Step(0)$. For *Eigenvalue ShatterPlots*, none of our extensive collections of graphs had the Shattering point missed. As seen in the *Eigenvalue* pattern presented in Section 6, all of our graphs are above the line, that is, E_s is higher than $1/\lambda_1 * E_t$ at the Shattering point. Therefore it is possible to overshot the initial value of ϵ to $1/\lambda_1$. If *Eigenvalue ShatterPlots* miss the Shattering point, an easy solution is to backtrack the algorithm and apply the *Proportional ShatterPlots* between 0 and the previous value of A_0. Wallclock times will be presented, illustrating the scalability of our method and the improvements reached with *Eigenvalue ShatterPlots*.

4 Experiments

This section presents the answers to our posed questions, our observations and the results achieved.

4.1 Datasets. Table 1 presents the symbols used in this section. The *Shattering point* is defined as the point where the shattering of the graph occurs. Based on this definition we will present the results of other measurements, such as nodes and edges of a giant component, total number

Symbols	Definitions
SP	Shattering point (= critical point)
REI	Random Edge insertion
ct	constant value
N_t	Total number of nodes in the graph
E_t	Total number of edges in the graph
Δ_t	Total number of Triangles in the graph
λ_1	Highest eigenvalue of original graph
N_s	Number of nodes at *SP*of degree ≥ 1
E_s	Number of edges at *SP*
d_s	Highest degree at *SP*
N_{sgcc}	Nodes in largest weakly conn. comp. at *SP*
E_{sgcc}	Edges in largest weakly conn. comp. at *SP*
$\lambda_{1,s}$	Highest eigenvalue at *SP*
Δ_s	Total number of Triangles at *SP*
D_s	Effective diameter at *SP*

Table 1: Symbols, acronyms and definitions

of reachable pairs, number of nodes, number of edges, diameter, highest degree, triangles and first eigenvalue at this point, named respectively N_{sgcc},E_{sgcc}, N_{Npairs}, N_s, E_s, D_s, d_s, Δ_S, and $\lambda_{1,s}$.

Table 2 presents all the datasets used and the symbols that represent each of them in the plots shown in the following sections. The synthetic datasets were generated using the algorithm described in their respective papers. For RB the model describing [40] was used with 3, 4 and 7 levels for each of the three graphs. For Erdős-Rényi the model used is presented in [16], but instead of $G(n, p)$, where p is the probability of attaching an edge, and n is the number of nodes, model $G(n, m)$ was preferred, where m is the total number of edges in a graph. The number of nodes and edges used are $n = 1k, 2k, 10k, 100k$ and $m = 5k, 14k, 50k, 400k$ respectively.

The Preferential Attachment graphs (PA) were created using the model described in [7] and using 3k and 4k as parameter of node and 3 and 5 as parameter of degree. In Small Word graphs (SW), the generator follows the model presented in [48] using the number of nodes (n), degree (d) and Rewire Probability (p) as parameters. Therefore, for the graphs in this paper we used $n = 5k, 8k, 8k$, $d = 5, 6, 3$ and $p = 0.4, 0.9, 0.5$, respectively. For 2D grids of 30x30, 50x50 and 1000x1000 were created without wrapping up. All of the graphs were undirected.

4.2 Choice of shattering criterion. Among the several measurements used to detect critical point/Shattering point, the best is the effective diameter D. The reason is that a giant component and a number of reachable pairs do show a critical point, that is, a sudden increase, as more and more edges are inserted, but it is not clear how to define

	Nodes	Edges	Description
Online social networks			
◇	75,877	405,739	Epinions network [42]
◇	33,696	180,811	Enron email net [28]
Academic collaboration (co-authorship) networks			
*	21,363	91,286	Arxiv `cond-mat` [33]
*	11,204	117,619	Arxiv `hep-ph` [33]
Information (citation) networks			
x	34,401	420,784	Arxiv `hep-th` citations [22]
x	32,384	315,713	Blog citation (1 year) [34]
Web graphs			
⊙	319,717	1,542,940	Stanford – UC Berkeley
⊙	855,802	4,291,352	Google web graph [1]
Amazon Product co-purchasing networks			
⊡	473,315	3,505,519	Snapshot 2 [13]
Bipartite (authors-to-papers) networks			
+	54,498	131,123	Arxiv `astro-ph` [34]
Internet networks			
■	13,579	37,448	AS Oregon [31]
■	22,963	48,436	AS graph from M. Newman
■	62,561	147,878	Gnutella, 31 Mar 2000 [43]
Grid networks			
♦	4,941	6,594	Power Grid western US [48]
Synthetic networks			
▽	2D - Synthetic Grid		
▼	Erdős-Rényi random graphs [18]		
▽	BR - Barabasi Hierarchical Model [40]		
▽	SW - SmallWorld [48]		
▽	PA - Preferential Attachment [7]		

Table 2: Datasets considered in our study. Their symbols at the beginning of each row are later used in figures to denote the datasets.

the exact Shattering point. In contrast, the diameter always has a sharp peak, reminding us of the percolation threshold [44]. Indeed the diameter is widely use to evaluate the network breakdown during the random node deletion or highest degree node deletion [4]. Figure 3 shows Gnutella, AS-Oregon and Author-to-Paper datasets - the others have been omitted for brevity, as they all have a similar behavior.

Rows correspond to measurements (diameter D_s, N_{sgcc} and N_{Npairs}). Each plot shows the measurement of interest (diameter, etc) versus the number of retained edges, under random edge insertion (*REI*).

The vertical lines correspond to the spike of the diameter (Top row plot) As seen in Figure 3, it is possible to use ShatterPlots to find the critical point, as it is the only one with

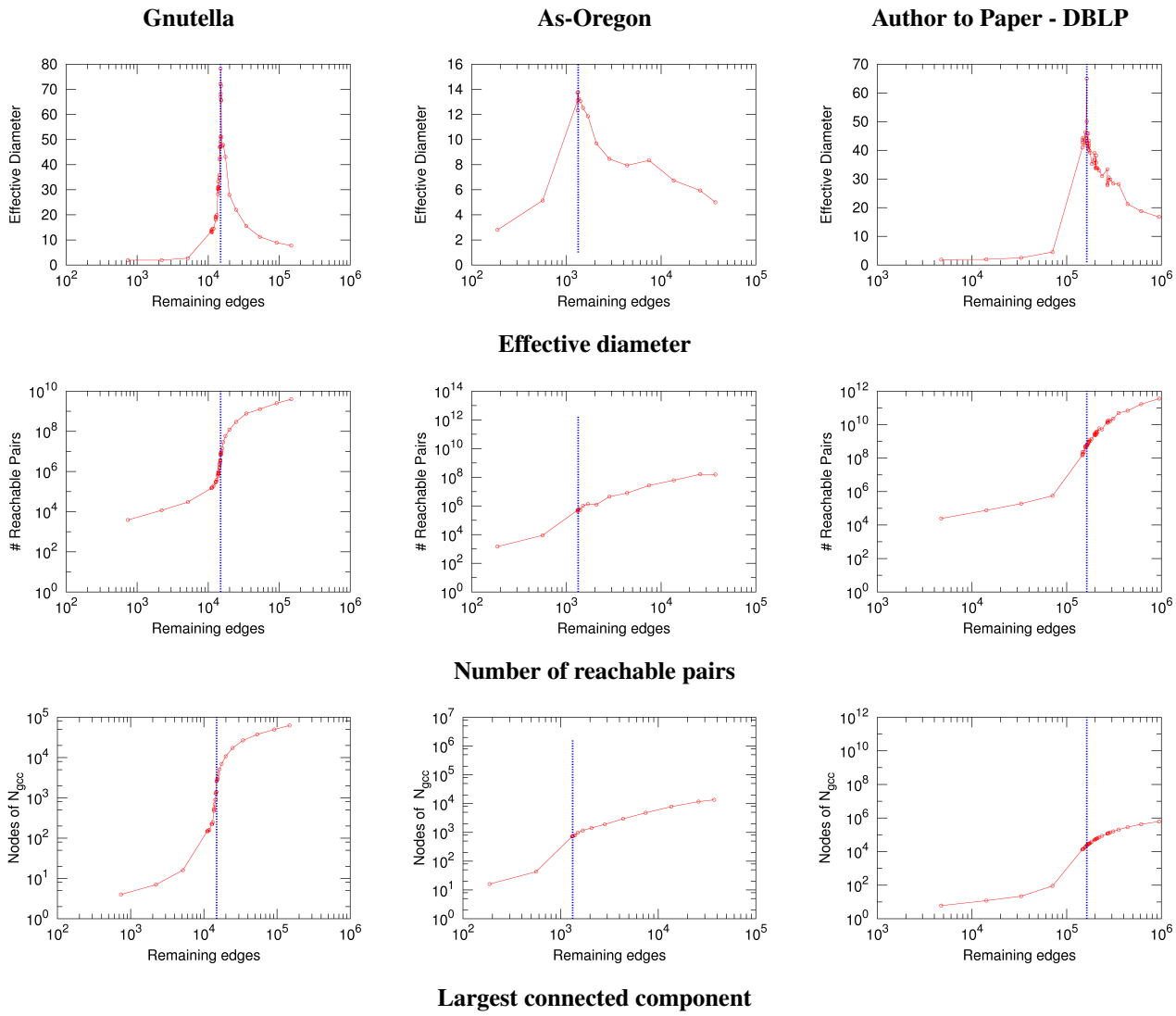


Figure 3: We randomly deleted edges and measurement graph structural properties. Graphs shatter in all measurement but only the diameter has a nice and clear spike.

a sharp, clear spike. The main point is that the Shattering point E_s, at which the diameter spikes, always fall in the region where the other measurement have a sudden drop. Now, we have:

DEFINITION 2. *The* Shattering point E_s *of a graph is the number of retained edges for which the (effective) diameter spikes.*

The ShatterPlot is exactly the plot of diameter D_s versus retained edges E_s. The remaining measurements shall not be used.

Another important definition is:

DEFINITION 3. *For Erdős-Rényi graphs, the Shattering point as defined above, coincides with the phase transition point.*

This is important, as there exist several results from the theory of random graphs used as sanity checks to our findings.

5 Results - Philosophical Questions

To answer the philosophical and exploratory questions posed in section 3.1 many graphs were "shattered" (Table 2 presents the datasets used) and plots were built with the measurements collected at Shattering point - N_s, E_s, d_s and $\lambda_{1,s}$ of all our real graphs. Such was also made for synthetic, Erdős-Rényi graphs, 2D-grid graphs, Hierarchical graphs, Small Word, and Preferential Attachment for verification and comparisons.

After "Shattering" real and synthetic graphs, it was

possible to answer both philosophical questions PhQ 1 and PhQ 2 based on the following patterns:

PATTERN 1. *All measurements have a Shattering point at about the same point for a given graph, but only the diameter has a clear spike.*

PATTERN 2. *All graphs have a Shattering point, under* REI.

Figure 4 shows the plots of structural measurements at the Shattering point. The axis scaling is linear - linear to (d), and log - log to (a), (b), (c), (e) and (f) and the theoretical/expected fitting curve (all of them with coefficient above 0.98), when there seems to be a strong correlation, are also shown. Moreover, the fitting lines are displayed - a blue one for the results we obtained, and a red one for the theoretical or expected ones. All experiments are average of 10 runs. The results for the Erdős-Rényi graphs are shown with dark down triangles, and the synthetic ones with down triangles for a better viewing in black-and-white. However, the paper is better viewed in color.

6 Results - Exploratory Questions

As seen in Figure 4 (a) all graphs have a Shattering point. The nodes-edges ratio at Shattering point N_s/E_s of all graphs follows a line which has a slope of 1.30, meaning that at the Shattering point the number of nodes N_s is about 30% higher than E_s. This observation also answered Question PhQ2 and part of ExQ3 and ExQ4.

When applied to Erdős-Rényi graphs, *REI* leads to a Shattering point which is exactly the one predicted by theory. In all our Erdős-Rényi graphs, the Shattering value E_s satisfied $E_s = N_t/2$ and $N_s = N_t * (1 - 1/e)$, which is exactly the condition for phase transition [16].

6.1 *30-per-cent* pattern.

PATTERN 3. (*30-per-cent*) *All real graphs shatter when N_s is about 30% higher than E_s.*

Theoretical Justification For Erdős-Rényi graphs, the *30-per-cent* pattern can be proved: For Erdős-Rényi graphs in the phase transition (= Shattering point), one has

$$E_s = 1/2 * N_s * e/(e-1) = 0.79 * N_s \tag{6.1}$$

where $e = 2.718$. Identically, $N_s = 1.26E_s$, which very close to 30%.

Proof. At Shattering point $N_s = N_t * (1 - 1/e)$ and $E_s = N_t/2$. Substituting N_t in the first equation by $2 * E_s$ the proof becomes complete. **QED**

Discussion: It is surprising that the remaining graphs also obey this pattern reasonably close. It is even more surprising, as further demonstrated at the Shattering point, real graphs clearly differ from Erdős-Rényi graphs, when aspects other than the E_s/N_s ratio (Question ExQ3) are considered.

Outliers: This is one of the few patterns that seems universal, and can not help us spot outliers and masked/synthetic graphs. Several of our upcoming patterns do, though.

6.2 *Eigenvalue* pattern.

Let E_s/E_t be defined as the *Edge Shattering Ratio*, which is the fraction of edges that needs to be retained to be at the Shattering point. Figure 4 (b) shows that the percentage of edges remaining in the graph at the Shattering point has a correlation with $1/\lambda$. This observation answered Question ExQ 1. Indeed, this pattern shows that the Edge Shattering Ratio does not depend on the size of the graph, but on the highest eigenvalue. Therefore one has:

PATTERN 4. (*Eigenvalue*) *The edges ratio*

$$E_s/E_t = ct * 1/\lambda_1. \tag{6.2}$$

Theoretical Justification: The Edge Shattering Ratio is the percentage of edges that still create a giant connected component. λ_1 is the epidemic threshold for an SIS model (Susceptible-Infected-Susceptible), like the flu virus:

THEOREM 6.1. *The epidemic threshold in an SIS model is $\beta/\delta = 1/\lambda_1$,*

where β is the virus birth rate, δ is the virus death rate and λ_1 is the highest eigenvalue of the original graph.

Proof. See [12] **QED**

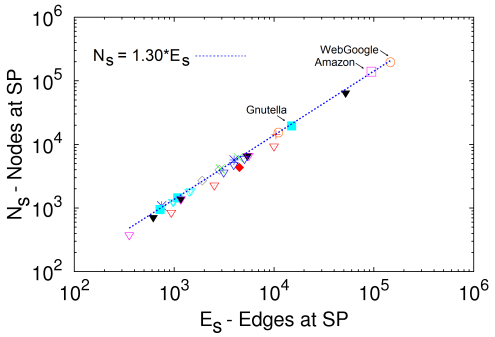
Discussion: β/δ is the number of attacks per edge that a virus-molecule can perform until the host has recovered. Thus, during the lifetime of a virus-molecule, it has $\delta \cong E_t$ edges available to it. At the epidemic threshold, this edge count should be $\beta \cong E_s$. The E_s/E_t ratio is also known as *Bond Percolation Threshold*. For 2D-grids the Bond Percolation Threshold is well defined as 0.5 [26].

Outliers: In this pattern one can observe that some graphs, like Preferential Attachment (PA), Hierarchical (RB) and 2D-grids stand out.

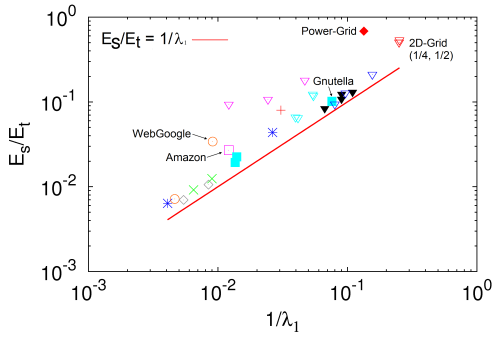
6.3 *NodeShatteringRatio* pattern.

Figure 4 (c) shows the Node Shattering Ratio, which is the relation of nodes at the Shattering point N_s (degree ≥ 1) versus number of nodes of the entire graph N_t. Three lines have been fitted in Figure 4 (c). Line (a) - dotted line - is exactly $N_t = N_s$, which is the maximum bound; line (b) - solid line - is the theoretical line of Erdős-Rényi and line (c) - dashed line - is $N_s = 0.37 * N_t$ below of which all real graphs are found. As we can see, this pattern answered Questions ExQ 2 and ExQ 3.

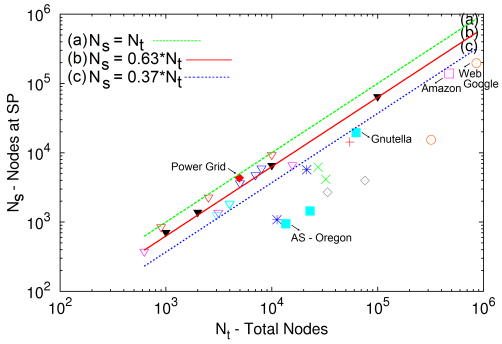
PATTERN 5. *Synthetic graphs are close to $N_s = 0.63 * N_t$.*



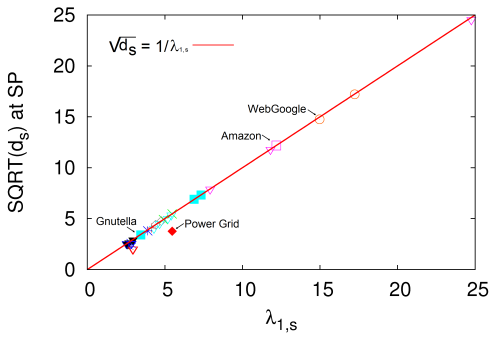
(a) N_s vs. E_s



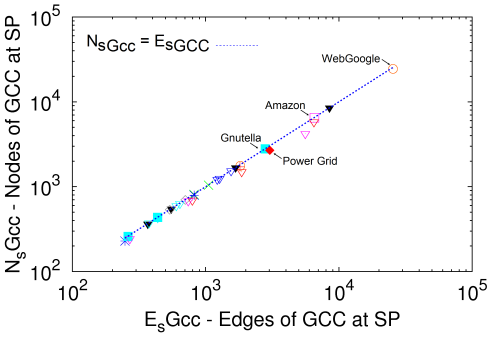
(b) E_s/E_t vs. $1/\lambda_1$



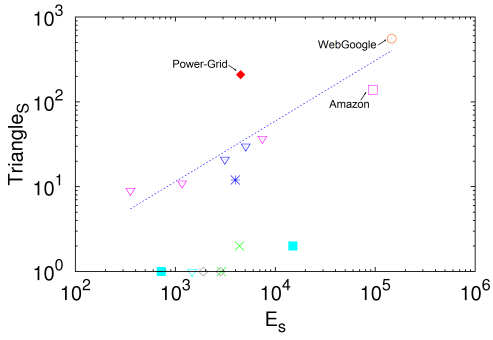
(c) N_s vs. N_t



(d) $\sqrt{d_s}$ vs. $\lambda_{1,s}$



(e) N_{sgcc} vs. E_{sgcc}



(f) Δ_s vs. E_s

Figure 4: Structural observations at the Shattering point (*SP*), where the graph shatters. Synthetic graphs in triangles; Erdős-Rényi ones in black triangles. (a) number of non-isolated nodes (N_s) versus number of edges E_s at the Shattering point (*30-per-cent* pattern); (b) number of retained edges E_s over total number of edges E_t versus one over λ_1, first eigenvalue of original graph (*Eigenvalue* pattern); Amazon is deviated from the line, as well as synthetic graphs RB, PA and 2D-grid. (c) number of survivor nodes N_s versus total number of nodes in the original graph N_t (*NodeShatteringRatio* pattern) 2D-grid are above the Erdős-Rényi line; SW are together with Erdős-Rényi , RB and PA are above line 'c' and below line 'b' (d)square root of highest degree at the Shattering point d_s versus $\lambda_{1,s}$ at Shattering point (*Root-degree* pattern); (e) number of nodes N_{sgcc} versus number of edges E_{sgcc} in the giant component at the Shattering point (*TreeGCC* pattern); (f) number of Triangles Δ_s versus number of edges E_s (*TriangleRatio* pattern). It is possible to observe that the Power Grid has a disproportionate number of triangles. Only the graphs with one or more triangles appear.

Theoretical Justification: As shown in [16], for all Erdős-Rényi in the phase transition one has

$$N_s = N_t * (1 - 1/e) \tag{6.3}$$

and $(1 - 1/e) = 0.63$, where $e = 2.718$.

Discussion: The explanation is that most real graphs have many nodes with degree $d = 1$, which is a heavy tail power law distribution, and these nodes have a high probability of being isolated at the Shattering point. An example is the AS Oregon dataset, in which the degree distribution is presented in 5 (c). On the other hand most nodes of graphs like 2D-grids have degree four, and Erdős-Rényi graphs have a little variation, with most nodes having their degree close to the average degree. All such graphs have very few isolated nodes when they shatter, with even fewer 2D-grids than Erdős-Rényi graphs. This is the reason why the orange triangles (2D-grids) are above the line of the black triangles (Erdős-Rényi graphs). In this way, this pattern shows that synthetic graphs have many more nodes at Shattering point than real ones. Although some graphs, like Amazon and Gnutella (as shown in Figures 5 (a) and (b)), are masked, they do not have a nice power law distribution. As seen in 4 (c) these graphs shatter faster than the other real graphs, like AS Oregon.

Outliers: The *NodeShatteringRatio* pattern is probably the best detector of synthetic and masked graphs, at least for the mix of graphs that have been studied in this paper. Notice that all synthetic graphs are close to line 'b' and above line 'c' - $N_s = 0.37 * N_t$ - in Figure 4(c).

6.4 *Root-degree* pattern. Figure 4 (d) plots the highest eigenvalue at the Shattering point $\lambda_{1,s}$, versus d_s, the square root of the highest degree in the graph at the Shattering point. The Figure also shows the line with equation $\lambda_{1,s} = \sqrt{d_s}$.

PATTERN 6. *All graphs obey* $\lambda_{1,s} \geq \sqrt{d_s}$.

Some recent theorems have helped us justify this behavior:

Theoretical Justification: As shown in [35] for all graphs, one has $\sqrt{d_i}(1 - o(1)) \leq \lambda_i \leq \sqrt{d_i}(1 + o(1)), i = 1, 2, ..., k$

where λ_i is the i-th eigenvalue and d_i is its respective degree.

Theoretical Justification: As shown in [17], for all Erdős-Rényi graphs one has $\lambda = [1 + o(1)] * \max(N * p, \sqrt{degree_{max}})$

where λ is the highest eigenvalue, N is the number of nodes of a graph, p is the probability that a node will be connected and $degree_{max}$ is the maximum degree of the graph.

Discussion: The theory presented above holds for any graph, including the ones at the Shattering point. At the Shattering point Erdős-Rényi graphs have $N*p$=1, given that the maximum degree will be > 1. Base on this assumption one can see why the pattern holds for Erdős-Rényi graphs.

Specifically for Erdős-Rényi graphs (black triangles), it is possible to observe that their eigenvalue $\lambda_{1,s}$ is roughly constant, independent of the number of nodes N_t the graph started with.

PATTERN 7. *The* $\lambda_{1,s}$ *for Erdős-Rényi graphs seems to be constant:* ≈ 2.8

the Power-Grid graph is below the line, meaning that it is well connected at the Shattering point. Figure 6 shows the highest degree node of the Power-Grid in the original graph (Figure 6 (a)) and at the Shattering point (Figure 6 (b)). It is possible to observe that the highest degree node still has some triangles and many connections even at the Shattering point. We can also verify it by looking at the *NodeShatteringRatio* pattern as the Power Grid is very close to line 'a' (Figure 4 (c)), that is, N_s is very close to N_t.

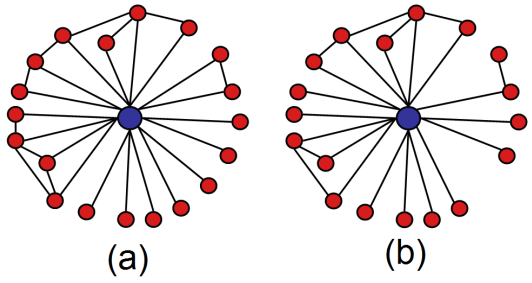
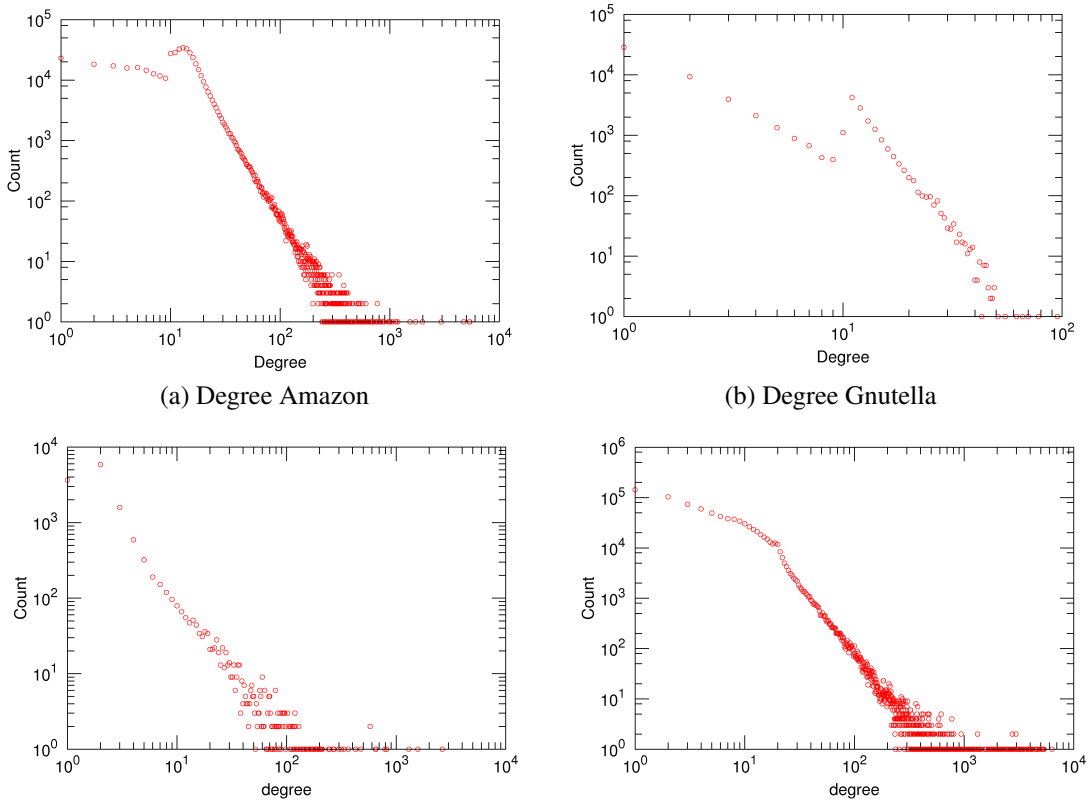


Figure 6: Highest degree node of Power-Grid: (a) Original Graph and (b) at the Shattering Point

6.5 *TreeGCC* pattern. Figure 4 (e) shows that all graphs at the Shattering point have the same amount of edges E_{sgcc} and nodes N_{sgcc} in the Giant Connected Component. As we can see, this pattern answered the second part of Question ExQ 4.

PATTERN 8. *All giant connected components of all graphs at the Shattering Point have* $E_{sgcc} \cong N_{sgcc}$.

Discussion: It is known that above the Shattering point the graph is well connected and below it the graph is completely disconnected. Therefore, at the Critical/Shattering point the graph is expected to be barely connected, meaning that a small amount of edges removed makes the graph totally disconnected. By observing this pattern, we can see that the Giant Connected Component at the Shattering Point looks like a tree. Notice that some graphs are plotted slightly below the line (apparently, being 'fatter' than a tree), for example Power Grid. Also notice the subtle difference between



(a) Degree Amazon

(b) Degree Gnutella

(c) Degree AS Oregon

(d) Degree Web Google

Figure 5: Degree Distribution of initial Graphs: (a) Amazon, (b) Gnutella, (c) AS Oregon and (d) Web Google graphs

this pattern and the *30-per-cent*: here the (several) nodes are ignored, and edges outside the giant connected component, while in the *30-per-cent* pattern, are included.

6.6 *TriangleRatio* pattern. Figure 4 (f) shows that, at the Shattering point, most of the graphs have very few triangles. In fact, the graphs with zero triangles were not ploted, due to the logarithmic axis.

PATTERN 9. *Graphs at the Shattering point have few or no triangles ($\Delta_s \approx 0$).*

Outliers: The Power Grid graph stands out.

Discussion: Graphs at the Shattering point are expected to be barely connected. We can see it in the *TreeGCC* pattern, where the giant connected component seems to be a tree, and in the *Root-degree* pattern, where $\lambda_{1,s}$ is strongly related to the highest degree at the Shattering point. We also know that the number of triangles (Δ) a node participates in increases according to the degree of that node [47]. However some graphs, like Power Grid, have a lot of triangles at the Shattering point. Why does the Power Grid exhibit such a different behavior? Some explanations are:

The Power Grid falls below the line in Figure 4 (d), which means that it has more edges than the nodes in the giant component, that is, the graph is "fatter" than a tree. Another fact is that $\lambda_{1,s}$ is higher than $\sqrt{d_s}$, as shown in Figure 4 (d), meaning that the eigenvalue is not correlated with the highest degree node, given that the highest degree node is better connected than a star, as shown in Figure 6 (b).

Another fact is that the relation between the initial number of triangles (Δ_t) of Power Grid is much higher than the other graphs. For example, initially, Power Grid has $\Delta_t = 651$ while Web Google has $\Delta_t = 13,356,298$; at the Shattering point, Power Grid has $\Delta_s = 209$ while Web Google has $\Delta_t = 556$.

7 Scalability

The ShatterPlots is a fast tool that needs to read the edge file only once at every iteration The number of iterations depends on how quickly we can zoom to the shattering point E_s.

Figure 7 shows the scalability of *Proportional ShatterPlots* and *Eigenvalue ShatterPlots*, plotting the wall-clock time versus the dataset size. The input graphs are synthetic Erdős-Rényi graphs, where the number of initial edges E =

14k, 40k, 50k, 200k, 300k, 500k, 600k and the number of nodes N=2k, 10K, 10k, 40K, 60k, 80k, 100k respectivily, were controled. The experiments ran on a Quad Xeon (2.66 GHz), with 8Gb of RAM, under Linux (Ubuntu).

Black and gray triangles correspond to the *Proportional ShatterPlots* and *Eigenvalue ShatterPlots* methods, respectively. The same datasets were used for both algorithms. The fitting lines (dotted-black, and solid red) show that both methods seem to scale up linearly with the graph size significantly faster than *Eigenvalue ShatterPlots* (up to 8x).

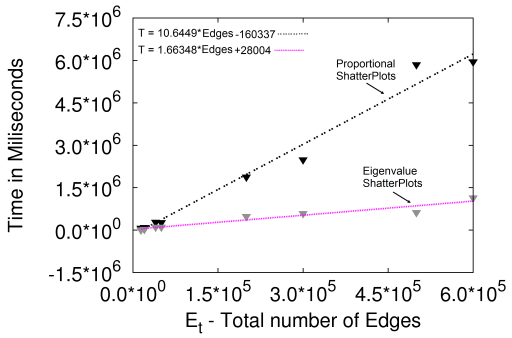


Figure 7: Scalability of *Proportional ShatterPlots* is represented by black double dotted line on dark triangles and *Eigenvalue ShatterPlots* by pink dotted line on gray triangles.

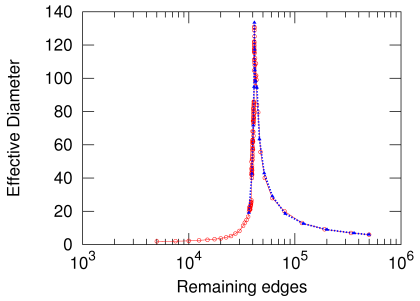


Figure 8: ShatterPlots of an Erdős-Rényi graph with 500k edges using *Eigenvalue ShatterPlots* (blue triangles) and *Proportional ShatterPlots* (red circles).

8 Conclusions

This paper proposed a new tool for studying graphs called 'ShatterPlots', and showed the surprising patterns found to help us spot masked and synthetic graphs. The main idea of 'ShatterPlots' is to use a "crash test" approach: we propose to shatter the graph, and observe its behavior. Our contributions are:

- A careful, scalable design of the tool. ShatterPlots needs less than $O(E)$ effort on each iteration, and a small number of iterations, due to our adaptive method.
- The use of *Eigenvalue* pattern to optimize the ShatterPlots (up to 8 times).
- Our observations, and confirmation/demolition of conjectures:
  - all criteria shatter at the same point, but only the diameter has a clear, sharp edge.
  - real graphs are far from the Shattering point
- Discovery of new patterns:
  - the Shattering point is at $1/\lambda_1 \cong E_s/E_t$, as one might expect from the epidemic threshold theory;
  - the *30-per-cent* pattern states that for all graphs used, at the Shattering point a graph has 30% more nodes than edges.
  - the *NodeShatteringRatio* pattern which allows separating real graphs from synthetic ones.
- Our patterns can spot synthetic/masked graphs

Future work could focus on the analysis of graphs over time, as well as on the parallelization of the method, say, on a 'hadoop'/map-reduce architecture.

Acknowledgements

Ana Paula Appel would like to acknowledge CAPES (PDEE project number 3960-07-2), CNPq and Fapesp for the financial support given to this research.

References

[1] Google programming contest. `http://www.google.com/programming-contest/`, 2002.

[2] C. Aggarwal and P. Yu. Outlier detection for high-dimensional data. In *SIGMOD*, pages 37–46, 2001.

[3] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 2002.

[4] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–381, 2000.

[5] N. T. J. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Hafner Press, 2nd edition, 1975.

[6] P. Bak. How nature works : The science of self-organized criticality, Sept. 1996.

[7] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, pages 509–512, 1999.

[8] V. Batagelj and M. Zaversnik. Generalized cores. *ArXiv*, (cs.DS/0202039), Feb 2002.

[9] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web: experiments and models. In *WWW Conf.*, 2000.

[10] J. M. Carlson and J. Doyle. Highly optimized tolerance: A mechanism for power laws in designed systems. *Physics Review E*, 60(2):1412–1427, 1999.

[11] D. Chakrabarti. AutoPart: Parameter-free graph partitioning and outlier detection. In *PKDD*, 2004.

[12] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos. Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur.*, 10(4):1–26, 2008.

[13] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks, August 2004.

[14] A. H. Dekker and B. D. Colbert. Network robustness and graph topology. In *ACSC '04: Proceedings of the 27th Australasian conference on Computer science*, pages 359–368, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[15] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 03)*, Washington, DC, August 24-27 2003.

[16] R. Durrett. *Random Graph Dynamics*. Cambridge University Press, Cambridge, 2007.

[17] A. Engel. On large deviation properties of erdos-renyi random graphs. *Journal of Statistical Physics*, 117:387–426(40), November 2004.

[18] P. Erdös and A. Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Acadamy of Science*, 5:17–61, 1960.

[19] A. Fabrikant, E. Koutsoupias, and C. H. Papadimitriou. Heuristically Optimized Trade-offs: A new paradigm for power laws in the Internet (extended abstract), 2002.

[20] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *SIGCOMM*, pages 251–262, 1999.

[21] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–71, 2002.

[22] J. Gehrke, P. Ginsparg, and J. Kleinberg. Overview of the 2003 KDD Cup. *SIGKDD Explorations*, 5(2):149–151, 2003.

[23] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proc. Natl. Acad. Sci. USA*, volume 99, 2002.

[24] R. Kannan, S. Vempala, and A. Vetta. On clusterings – good, bad and spectral. In *FOCS*, 2000.

[25] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48:96–129, 1998.

[26] H. Kesten. The critical probability of bond percolation on the square lattice equals 1/2. *Communications in Mathematical Physics*, 74:41–59, Feb. 1980.

[27] J. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. In *Proceedings of the International Conference on Combinatorics and Computing*, 1999.

[28] B. Klimt and Y. Yang. Introducing the enron corpus. In *CEAS*, 2004.

[29] J. S. Kong, B. A. Rezaei, N. Sarshar, V. P. Roychowdhury, and P. O. Boykin. Collaborative spam filtering using e-mail networks. *Computer*, 39(8):67–73, 2006.

[30] R. Kumar, A. Tomkins, and E. Vee. Connectivity structure of bipartite graphs via the knc-plot. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 129–138, New York, NY, USA, 2008. ACM.

[31] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):2, 2007.

[32] J. Leskovec, J. M. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD '05: Proceeding of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, 2005.

[33] J. Leskovec, J. M. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.

[34] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SDM '07: Proceedings of the XXXth SIAM Conference on Data Mining*, 2007.

[35] M. Mihail and C. Papadimitriou. On the eigenvalue power law. In *RANDOM*, Harvard, MA, 2002.

[36] S. Milgram. The small-world problem. *Psychology Today*, 2:60–67, 1967.

[37] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *KDD*, pages 631–636, 2003.

[38] C. R. Palmer, P. B. Gibbons, and C. Faloutsos. ANF: A fast and scalable tool for data mining in massive graphs. In *SIGKDD*, Edmonton, AB, Canada, 2002.

[39] D. M. Pennock, G. W. Flake, S. Lawrence, E. J. Glover, and C. L. Giles. Winners don't take all: Characterizing the competition for links on the Web. *Proceedings of the National Academy of Sciences*, 99(8):5207–5211, 2002.

[40] E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Physical Review E*, September 2002.

[41] S. Redner. How popular is your paper? an empirical study of the citation distribution. *European Physical Journal B*, 4:131–134, 1998.

[42] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *ISWC*, 2003.

[43] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6(1), 2002.

[44] M. Schroeder. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. W.H. Freeman and Company, New York, 1991.

[45] R. Sole and B. Goodwin. *Signs of Life: How Complexity Pervades Biology*. Perseus Books Group, New York, NY, 2000.

[46] S. L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the Internet topology. In *Global Internet, San Antonio, Texas*, 2001.

[47] C. Tsourakakis. Fast counting of triangles in large real networks, without counting: Algorithms and laws. In *ICDM*, 2008.

[48] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.