Regularities and dynamics in bisimulation reductions of big graphs

Yongming Luo
Eindhoven University of Technology
y.luo@tue.nl

George H. L. Fletcher
Eindhoven University of Technology
g.h.l.fletcher@tue.nl

Jan Hidders
Delft University of Technology
a.j.h.hidders@tudelft.nl

Paul De Bra
Eindhoven University of Technology
debra@win.tue.nl

Yuqing Wu
Indiana University, Bloomington
yuqwu@cs.indiana.edu

ABSTRACT

Bisimulation is a basic graph reduction operation, which plays a key role in a wide range of graph analytical applications. While there are many algorithms dedicated to computing bisimulation results, to our knowledge, little work has been done to analyze the results themselves. Since data properties such as skew can greatly influence the performances of data-intensive tasks, the lack of such insight leads to inefficient algorithm and system design.

In this paper we take a close look into various aspects of bisimulation results on big graphs, from both real-world scenarios and synthetic graph generators, with graph size varying from 1 million to 1 billion edges. We make the following observations: (1) A certain degree of regularity exists in real-world graphs' bisimulation results. Specifically, power-law distributions appear in many of the results' properties. (2) Synthetic graphs fail to fulfill one or more of these regularities that are revealed in the real-world graphs. (3) By examining a growing social network graph (Flickr-Grow), we see that the corresponding bisimulation partition relation graph grows as well, but the growth is stable with respect to the original graph.

1. INTRODUCTION

Graphs have long been a fundamental data model in mathematics and computer science. Recently, with the proliferation of available graph data and the pressing need for graph analytics, massive graph management problems have been receiving increasing attention from the data management, semantic web and many other research communities. Graphs of interest, such as social networks [30], internet graphs [10] and linked open data [15], are on the order of millions or even billions of nodes and edges. To enable efficient analytics on such huge graphs, often one of the first tasks to perform is to adopt some graph reduction technique to shrink the size of the graphs, while still maintaining certain characteristics (e.g., topological structure).

Graph bisimulation partitioning (and its many variants) is such a reduction operation. Intuitively, bisimulation partitioning groups nodes together as disjoint sets based on the local topology of each node. These partition "blocks" and the relationships between them form an *abstracted* graph where the graph size is reduced but the structural information (e.g., path information) is preserved. With the help of such abstracted graph, many queries can be answered or filtered out without probing the real graph, therefore the graph management system's performance is greatly enhanced.

Bisimulation is a ubiquitous notion across many fields [27]. In the context of graph reduction, graph bisimulation finds its applications in various data management problems, such as constructing structural indexes for XML and RDF databases [13, 23, 25], graph compression [7, 12], and subgraph matching [11].

Inspired by numerous real-world applications, including those in data management, algorithms for computing bisimulation reductions have been studied for decades. Algorithms targeting various constraints and computational models, such as main-memory algorithms [1], I/O efficient algorithms [16, 22] and distributed solutions [6, 21], have been developed to efficiently compute bisimulation partitions of massive graphs. For example, the state-of-the-art MapReduce-based algorithm [21] can compute a "k-localized" variant of bisimulation, discussed below, on a social graph with 1.4 billion edges in a few hours, for $k = 10$.

Despite all of the aforementioned efforts, little work has been carried out to take a deep look into the bisimulation result itself, which is essential for applications (e.g., indexing, query optimization, compression, load balancing) to take into consideration. Indeed, it is well known that graph properties, or data properties in general, such as skewness (e.g., power-law distribution [8]) can hugely influence the performance of data-intensive processing. This applies to both single-machine algorithms (e.g., caching effects [9]) and distributed algorithms (e.g., [3, 18]). Therefore, characteristics of the input data must be examined and reflected at the stage of algorithm design.

Motivated by these observations, in this paper we analyze the localized k-bisimulation partitioning results of many real and synthetic big graphs. We focus on a localized variant of bisimulation due to its practical applicability in data management solutions (e.g., [13, 19]). We compare the graph properties of the *abstracted bisimulation graph* (defined as k-BPR graph in Def. 2) both with each other and with the original underlying graph. We also analyze a dynamic social network graph (Flickr-Grow), examining the behavior of the k-BPR graph as the original graph grows.

We make the followings observations:

- Regularities exist in the bisimulation results of real-world graphs. Power-law distributions hold for partition block size distribution, signature length distribution, degree distributions for the k-BPR graph. The k-BPR graphs are usually denser than their original graphs.
- In the context of bisimulation results, the synthetic graph generators that we examined fail to fulfill one or more of the regularities that are observed in real-world graphs.
- For the dynamic social network that we examined, its k-BPR graph also grows, but the growth is stable (related by a constant factor) with respect to the original graph.

To the best of our knowledge, we are the first to make these observations.

The rest of the paper is organized as follows. In Section 2 we introduce the basic definitions and notions we will use in the paper, as well as the experiment setup. In Section 3, we examine the bisimulation properties of static graphs. In Section 4, we further investigate on the behaviors of a growing social graph. We conclude in Section 5 with a discussion of future directions for research.

2. PRELIMINARIES AND EXPERIMENT SETUP

For a directed node- and edge-labeled graph $G = \langle N, E, \lambda_N, \lambda_E \rangle$, where N is a finite set of nodes, $E \subseteq N \times N$ is a set of edges, λ_N is a function from N to a set of node labels $\mathcal{L}_N$, and λ_E is a function from E to a set of edge labels $\mathcal{L}_E$, we define k-*bisimilar* equivalence relation on N:

Definition 1. *Let $G = \langle N, E, \lambda_N, \lambda_E \rangle$ be a graph and $k \geq 0$. Nodes $u, v \in N$ are called* k-bisimilar *(denoted as $u \approx^k v$), iff the following holds:*

1. *$\lambda_N(u) = \lambda_N(v)$,*
2. *if $k > 0$, then for any edge $(u, u') \in E$, there exists an edge $(v, v') \in E$, such that $u' \approx^{k-1} v'$ and $\lambda_E(u, u') = \lambda_E(v, v')$, and*
3. *if $k > 0$, then for any edge $(v, v') \in E$, there exists an edge $(u, u') \in E$, such that $v' \approx^{k-1} u'$ and $\lambda_E(v, v') = \lambda_E(u, u')$.*

Then we define the k-*bisimulation partition relation graph* from the $\approx^k$ relation.

Definition 2. *Let $G = \langle N, E, \lambda_N, \lambda_E \rangle$ be a graph and $k \geq 0$. The* k-bisimulation partition relation graph *for G (denoted as* k-BPR *graph) is the directed graph $G_k = \langle N_k, E_k \rangle$, such that*

- *N_k consists of the equivalence classes of $\approx^k$, i.e., if for node $v \in N$, we let $[v]_{\approx^k} = \{u \in N \mid v \approx^k u\}$, then $N_k = \{[v]_{\approx^k} \mid v \in N\}$.*
- *$E_k \subseteq N_k \times N_k$, and $(X, Y) \in E_k$ iff $\exists x \in X, y \in Y$ s.t. $(x, y) \in E$.*

Since both G and G_k are directed graphs, we define for each node in G and G_k the in-degree (out-degree) as the number of incoming (outgoing) edges of that node.

Example. Consider a simple directed graph in Figure 1, where edges indicate the "following" relationship in a social network. Nodes {1, 2, 3, 6} are labeled with "public" (filled with black), while {4, 5, 7, 8} are labeled with "celebrity" (filled with white). Under 1-bisimulation, we have that these nodes are partitioned into $P1 = \{1, 6\}$, $P2 = \{2, 3\}$, $P3 = \{4, 7\}$ and $P4 = \{5, 8\}$. As a result, the 1-BPR graph has edge set $\{P1 \rightarrow P1, P2 \rightarrow P2, P2 \leftrightarrow P1 \rightarrow P3 \rightarrow P4\}$. We see that the graph size is greatly reduced, while the 1-step node reachability information is maintained. For example, if we want to query the descendent nodes of node 8, since node 8 belongs to $P4$, we directly know that there is no outgoing edge from $P4$ (and therefore from node 8). Hence, there is no need to probe the original graph, and an empty set is immediately returned.

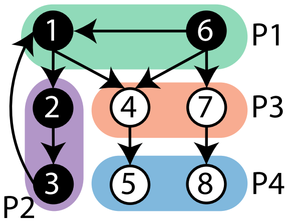


Figure 1: An example social network ("following" relation), different colors group nodes to different 1-bisimulation partition blocks

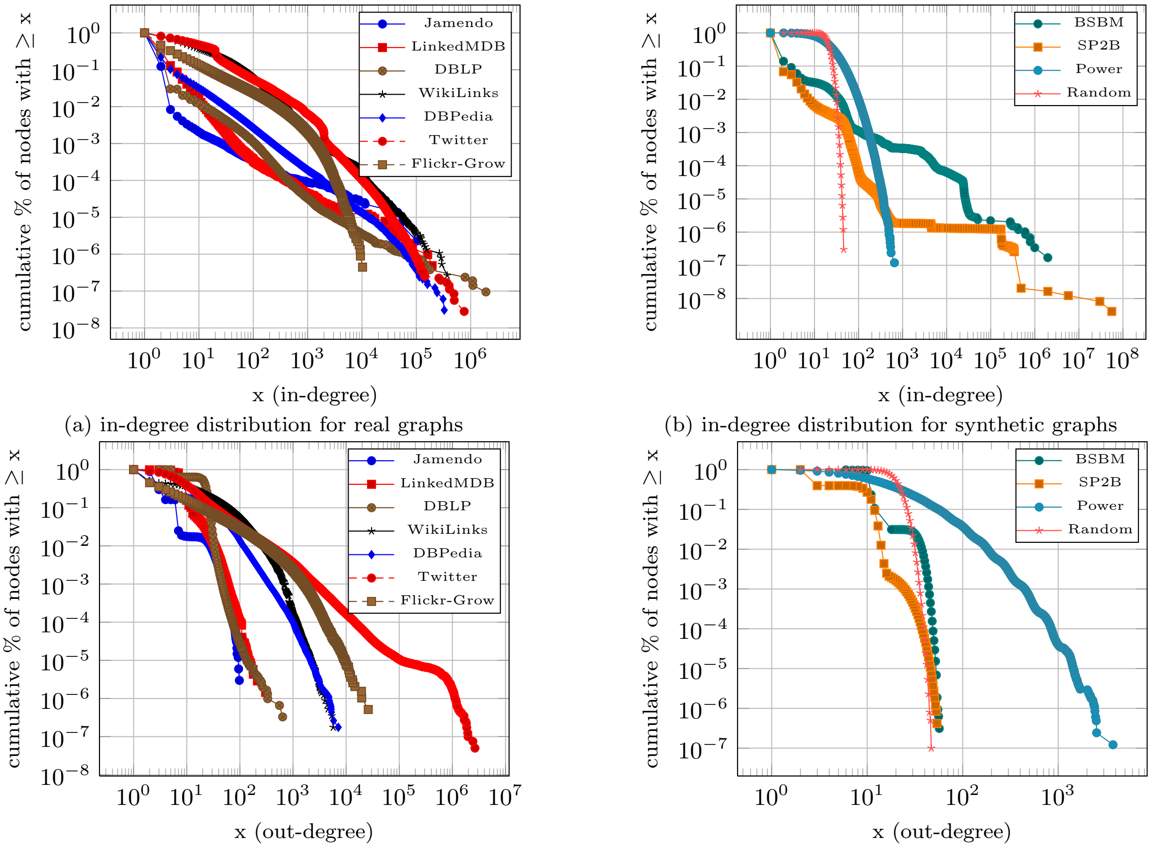
We refer the readers to Luo et al. [22] for a more detailed discussion of localized bisimulation. In the sequel, we consider k-BPR graphs with self-loops on nodes, though the difference is not significant. Also when we plot distributions for some graph properties, we use cumulative distribution function (CDF) [2]. Intuitively, CDF describes for some value x, the percentage of occurrences of samples with a value less than or equal to x.

Experiment setup. In this paper we use a state-of-the-art external memory algorithm [22] to compute the localized bisimulation result for all graphs. This algorithm enables us to process huge graphs with up to billions of edges. All experiments are executed on a cluster machine (Intel Xeon 2.27 GHz processor, 12GB main memory, Fedora 14 64-bit Linux). We compute to $k = 10$ since this is big enough to show all properties of interest from the k-*bisimulation* results.

Graph datasets. The graph datasets we use in this paper are collected from a wide range of applications. In Table 1 we show some simple statistics of the datasets. Figure 2 presents the in-degree and out-degree distributions for the real graphs and synthetic graphs respectively. We see all the real graphs and some synthetic graphs (i.e. BSBM, SP2B, Power) show a certain power-law distribution. For Flickr-Grow we plot the grown graph.

3. STATIC PROPERTIES OF k-BPR GRAPHS

In this section we examine the properties of the static graphs (we treat the grown Flickr-Grow as a static graph in this section). Specifically, we are interested in the comparison of basic structural properties of the k-BPR graph G_k



(a) in-degree distribution for real graphs

(b) in-degree distribution for synthetic graphs

(c) out-degree distribution for real graphs

(d) in-degree distribution for synthetic graphs

Figure 2: In-degree and out-degree distributions for graphs

Table 1: Description and statistics of the graph datasets

Data Name	Description	$\|N\|$	$\|E\|$	$\frac{\|E\|}{\|N\|}$
Jamendo (**E**)	A repository of music metadata in RDF format [26]	0.49M	1.05M	2.16
LinkedMDB (**E**)	A repository of movie metadata in RDF format [14]	2.33M	6.15M	2.64
DBLP (**E**)	An RDF format DBLP dump[1]	23M	50.2M	2.18
WikiLinks (**NO**)	A page-to-page linking graph of Wikipedia[2]	5.71M	130.16M	22.79
DBPedia (**E**)	An early RDF dump of DBPedia[3]	38.62M	115.3M	2.99
Twitter (**NO**)	A following relationship graph of Twitter [20]	41.65M	1468.37M	35.25
SP2B (**E**)	A RDF data generator for arbitrarily large DBLP-like data [29]	280.91M	500M	1.78
BSBM (**E**)	A RDF data generator for e-commerce use case [5]	8.89M	34.87M	3.92
Random (**E**)	Random graph generated by GTgraph [4]	10M	200M	20
Power (**E**)	Power-law distribution graph generated by GTgraph [4]	8.39M	200M	23.85
Flickr-Grow (**NO**)	A following relationship graph of Flickr [24]	1.5M to 2.3M	17.7M to 33.1M	11.68 to 14.39

* **E**, **N** and **NO** indicate the graph is labeled on edge, node or neither, resp.

and its original graph G.

3.1 Comparison of $\mathbf{G_k}$ and G

In Figure 3a and 3b we show $\frac{|N_k|}{|N|}$ and $\frac{|E_k|}{|E|}$ for $k \in \{1, \ldots, 10\}$ for all graphs, where $|X|$ denotes the size of set X. The figures indicate the reduction (compression) rate we can get. In general, we see that localized bisimulation reduction provides good compression on the original graphs, with reduction rate between 10^{-4} and 10^{-1}, and the rate becomes stable around $k = 5$. We also see that, compared with the real graphs, the partition results from synthetic datasets {BSBM, Power, Random} are either too coarse or too refined. However, this also happens for the real graphs without labels (i.e., WikiLinks, Twitter, Flickr-Grow).

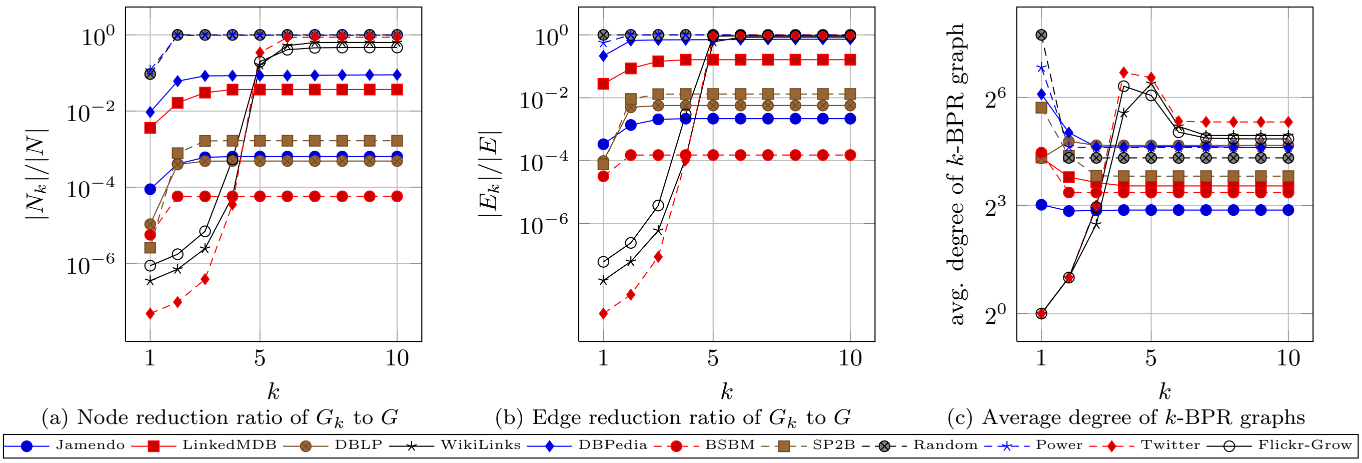
In Figure 3c, we plot the average degree of the partition graph for each dataset for $k \in \{1, \ldots, 10\}$. Comparing with the original graph degree in Table 1, we see that the partition block graphs usually have higher degrees, and, at the beginning of the computation, the average degree tends to drop. In the case of graphs without labels, the degrees first rise until k is 4 or 5 and then drop.

Overall, for the purpose of compression or structural indexing, we observe that choosing $k = 5$ is usually sufficient. A larger k value would lead to a too refined partitioning. k-BPR graphs are usually denser than their original graphs.

[1] http://thedatahub.org/dataset/l3s-dblp

[2] http://haselgrove.id.au/wikipedia.htm

[3] http://www.cs.vu.nl/~pmika/swc/btc.html



(a) Node reduction ratio of G_k to G (b) Edge reduction ratio of G_k to G (c) Average degree of k-BPR graphs

Figure 3: Comparison of k-BPR graph to its original graph

3.2 Power-law distribution in G_k

In Section 2, we see that many of the original graphs follow a power-law distribution in their structure. We are curious about whether this is also true for their k-BPR graphs. The investigation can be found in Figure 4.

Figure 4a and 4b show the distribution of partition block size for each graph. Here note that for the Random dataset, each node belongs to its own partition.

Luo et al. [22] define a notion of *signature* for each node, which is essentially an encoding of the bisimulation equivalence class of the node. The length of a node's signature gives us insight into the complexity of the local topology of the node. Figure 4c and 4d show the distribution of signature lengths.

It would be interesting to further study some graph properties of the k-BPR graphs. In Figure 4e, 4f, 4g and 4h, we plot the in-degree and out-degree of the k-BPR graphs for real graphs and synthetic graphs, respectively.

In general, we observe that all examined properties show certain power-law distribution nature for real graphs. This gives us some insights when we want to build applications of k-BPR graphs. Furthermore, we note that not a single synthetic dataset fulfils all power-law distribution graph properties as shown in real data. From the bisimulation partition perspective, the most *real* synthetic graph is SP2B, which still, lacks of the power-law distribution on signature length. This indicates that benchmark graph generators still need to be improved in this direction to reflect the structure of real graphs.

4. DYNAMIC PROPERTIES OF k-BPR GRAPHS

While Section 3 studies the properties for static graphs and their k-BPR graphs, in this section we want to look into growing graphs. Note that for our growing graph (Flickr-Grow), the findings in Section 3 still hold.

It is easy to design synthetic graphs such that their corresponding k-BPR graph either shrinks or grows, as the original graph grows. For real-world social graphs, however, we are interested to know (**Q1**) is the k-BPR graph growing when the original graph grows?; and, (**Q2**) is the k-BPR graph growing faster than the original graph? We use the Flickr-Grow graph for this investigation. The original Flickr-Grow graph includes a time stamp for each edge. We separate the edge set into 14 subsets based on the time stamp, grouping edges together for every 10 days. In this way, we can examine graph growth in a coarse granularity.

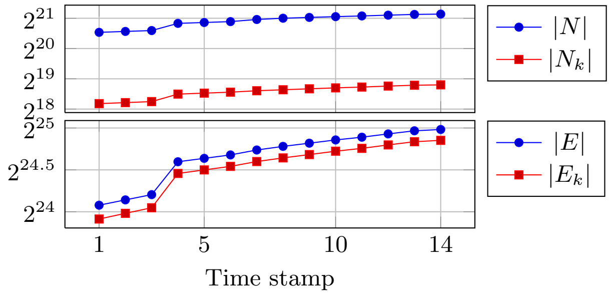


Figure 5: k-BPR graph growth trend in $|N|$, $|E|$, $|N_k|$ and $|E_k|$

To answer Q1, we plot in Figure 5 the trend of $|N|$ and $|E|$ of G, $|N_k|$ and $|E_k|$ of G_k with time, where $k = 5$. Other k values show the same behavior as well. Essentially, we examine the k-BPR graph growth in terms of nodes and edges. We see that during the whole period, $|N_k|$ increased by 1.5× and $|E_k|$ by 2×, while the original graph grows with the same ratios.

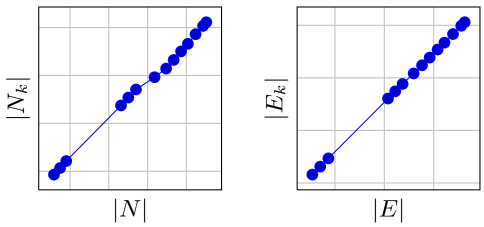
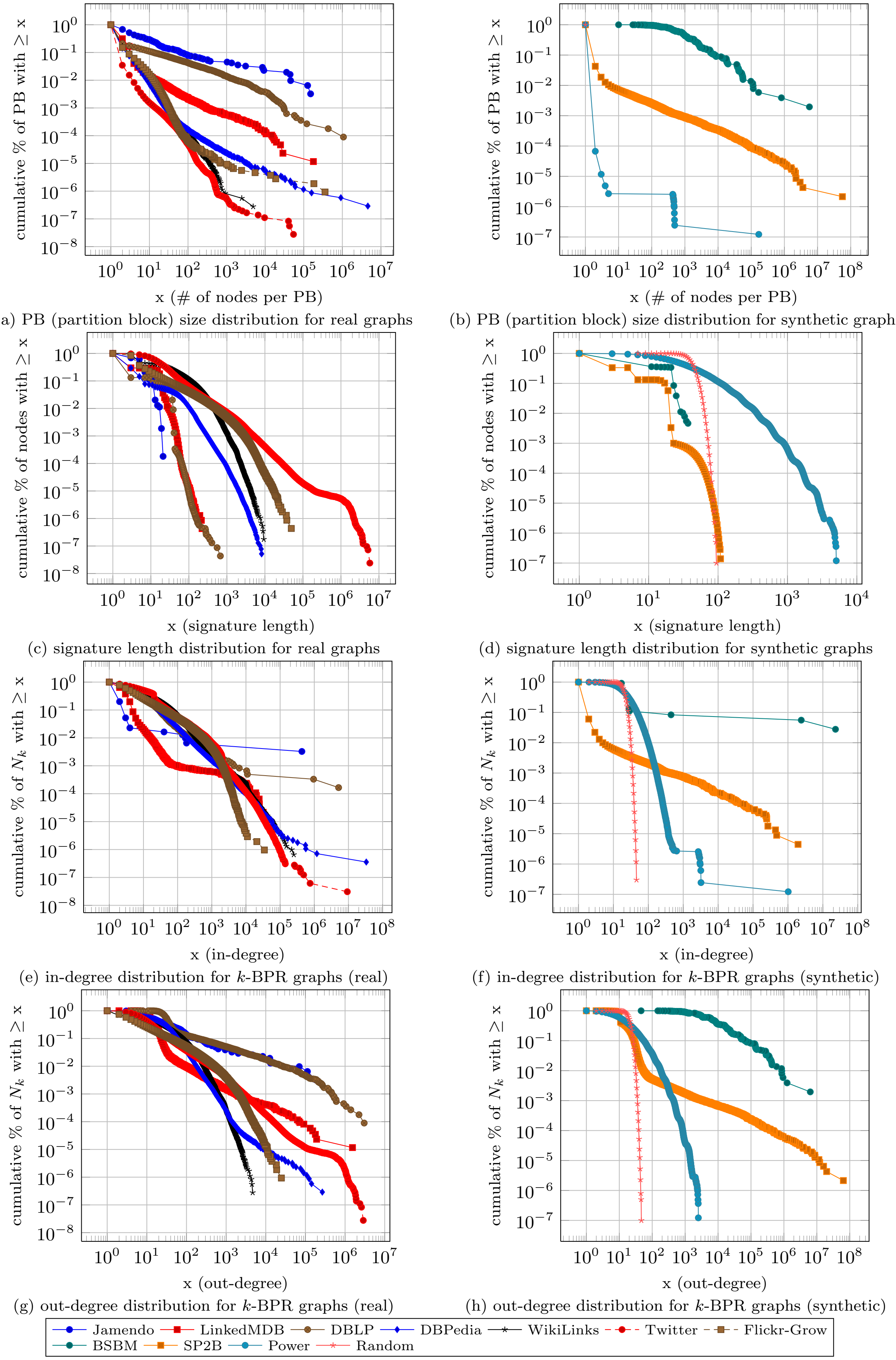


Figure 6: k-BPR graph growth trend in $|N_k|$ w.r.t. $|N|$ and $|E_k|$ to $|E|$, all axes are in linear scale

To answer Q2, we plot Figure 6, showing the growth of $|N_k|$ (y-axis) w.r.t. $|N|$ (x-axis) and $|E_k|$ to $|E|$. We see that there is clearly a constant factor between $|N_k|$ and $|N|$ ($|E_k|$ and $|E|$). So we conclude that (1) the k-BPR graph grows with the original graph, but (2) the growth is stable with respect to the original graph.



(a) PB (partition block) size distribution for real graphs

(b) PB (partition block) size distribution for synthetic graphs

(c) signature length distribution for real graphs

(d) signature length distribution for synthetic graphs

(e) in-degree distribution for k-BPR graphs (real)

(f) in-degree distribution for k-BPR graphs (synthetic)

(g) out-degree distribution for k-BPR graphs (real)

(h) out-degree distribution for k-BPR graphs (synthetic)

Figure 4: Distributions in k-bisimulation results

5. CONCLUSION AND DISCUSSION

In this paper, we have examined many aspects of the localized bisimulation partitioning results for massive real-world and synthetic graphs. Extensive experiments have shown basic regularities in the k-BPR graphs for both static and dynamic real graphs, while the synthetic graphs fail to mimic real graphs in this respect. To our knowledge, we are the first to make these observations.

Observations in this paper not only provide insight into other applications, as suggested above, but also provide directions for future research. First, other interesting measurements on the k-BPR graphs can be performed; features such as diameter and clustering coefficient may show different properties when compared with the original graphs. Second, it would be interesting to analyze the different behaviors of labeled and unlabeled graphs (as in Sec. 3), and determining the causes. Third, as we have seen throughout the paper, synthetic graph generators fail to deliver power-law distribution bisimulation results as observed in real graphs. Studying ways to solve this problem on existing graph generation models or with new models is an important research direction. Last but not least, similar research could be carried out on other related reductions, such as simulation partition graphs [17].

Acknowledgments. The research of YW is supported by Research Foundation Flanders (FWO) during her sabbatical visit to Hasselt University, Belgium. The research of YL, GF, JH and PD is supported by the Netherlands Organisation for Scientific Research (NWO).

6. REFERENCES

[1] L. Aceto, A. Ingolfsdottir, and J. Srba. The algorithmics of bisimilarity. In Sangiorgi and Rutten [28], pages 100–172.

[2] L. A. Adamic. Zipf, Power-laws, and Pareto - a ranking tutorial. `http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html`.

[3] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Phys. Rev. E*, 64:046135, Sep 2001.

[4] D. A. Bader and K. Madduri. GTgraph: A suite of synthetic graph generators. `http://www.cse.psu.edu/~madduri/software/GTgraph/index.html`.

[5] C. Bizer and A. Schultz. The Berlin SPARQL Benchmark. *IJSWIS*, 5(2):1–24, 2009.

[6] S. Blom and S. Orzan. A distributed algorithm for strong bisimulation reduction of state spaces. *Int J Softw Tools Technol Transfer*, 7:74–86, 2005.

[7] P. Buneman, M. Grohe, and C. Koch. Path queries on compressed XML. In *Proc. VLDB*, pages 141–152, Berlin, Germany, 2003.

[8] A. Clauset, C. Shalizi, and M. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[9] E. D. Demaine. Cache-oblivious algorithms and data structures. Lecture Notes from the EEF Summer School on Massive Data Sets. University of Aarhus, Denmark, June 27–July 1, 2002.

[10] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262, Aug. 1999.

[11] W. Fan. Graph pattern matching revised for social network analysis. In *Proc. ICDT*, pages 8–21, Berlin, Germany, 2012.

[12] W. Fan, J. Li, X. Wang, and Y. Wu. Query preserving graph compression. In *Proc. SIGMOD*, pages 157–168, Scottsdale, AZ, USA, 2012.

[13] G. H. L. Fletcher, D. Van Gucht, Y. Wu, M. Gyssens, S. Brenes, and J. Paredaens. A methodology for coupling fragments of XPath with structural indexes for XML documents. *Inf. Syst.*, 34(7):657–670, 2009.

[14] O. Hassanzadeh and M. P. Consens. Linked Movie Data Base. In *Proc. LDOW*, Madrid, Spain, 2009.

[15] T. Heath and C. Bizer. Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1):1–136, 2011.

[16] J. Hellings, G. H. L. Fletcher, and H. Haverkort. Efficient external-memory bisimulation on DAGs. In *Proc. SIGMOD*, pages 553–564, Scottsdale, AZ, USA, 2012.

[17] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke. Computing simulations on finite and infinite graphs. In *Proc. FOCS*, pages 453–462, Washington, DC, USA, 1995.

[18] K. A. Hua and C. Lee. Handling Data Skew in Multiprocessor Database Computers Using Partition Tuning. In *Proc. VLDB*, pages 525–535, San Francisco, CA, USA, 1991.

[19] R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes. Exploiting local similarity for indexing paths in graph-structured data. In *Proc. ICDE*, pages 129–140, San Jose, CA, USA, 2002.

[20] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proc. WWW*, pages 591–600, New York, NY, USA, 2010.

[21] Y. Luo, Y. de Lange, G. H. L. Fletcher, P. De Bra, J. Hidders, and Y. Wu. Bisimulation Reduction of Big Graphs on MapReduce. In *Proc. BNCOD*, Oxford, UK, to appear 2013.

[22] Y. Luo, G. H. L. Fletcher, J. Hidders, Y. Wu, and P. De Bra. I/O-efficient algorithms for localized bisimulation partition construction and maintenance on massive graphs. *CoRR*, abs/1210.0748, 2012.

[23] T. Milo and D. Suciu. Index Structures for Path Expressions. In *Proc. ICDT*, pages 277–295, Jerusalem, Israel, 1999.

[24] A. Mislove, H. S. Koppula, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Growth of the Flickr Social Network. In *Proc. WOSN*, Seattle, WA, USA, August 2008.

[25] F. Picalausa, Y. Luo, G. H. L. Fletcher, J. Hidders, and S. Vansummeren. A Structural Approach to Indexing Triples. In *Proc. ESWC*, pages 406–421, Heraklion, Greece, 2012.

[26] Y. Raimond, M. B. Sandler, and Q. Mary. A Web of Musical Information. In *ISMIR*, pages 263–268, Philadelphia, PA, USA, 2008.

[27] D. Sangiorgi. Origins of bisimulation and coinduction. In Sangiorgi and Rutten [28], pages 1–37.

[28] D. Sangiorgi and J. Rutten, editors. *Advanced Topics in Bisimulation and Coinduction.* Cambridge University Press, Cambridge, 2011.

[29] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel. SP^{2}Bench: A SPARQL Performance Benchmark. In *Proc. ICDE*, pages 222–233, Washington, DC, USA, 2009.

[30] J. Scott. *Social network analysis.* SAGE Publications Limited, 2012.