Distributed and Streaming Graph Processing Techniques

Panagiotis Liakos*

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
p.liakos@di.uoa.gr

Abstract. Beneath most complex systems playing a vital role in our daily lives lie intricate networks. Such real-world networks are routinely represented using graphs. The volume of graph data produced in today's interlinked world allows for realizing numerous fascinating applications but also poses important challenges. Consider for example the friendship graph of a social networking site and the findings we can come up with when executing network algorithms, such as community detection, on this graph. However, the volume that real-world networks reach oftentimes makes even the execution of fundamental graph algorithms infeasible when following traditional techniques. In this short note we summarize our results on the study of two directions that allow for handling large scale networks, namely *distributed graph processing*, and *streaming graph algorithms*. In this context, we first provide contributions with regard to memory usage of distributed graph processing systems by extending the available structures of a contemporary such system with memory-optimized representations. Then, we focus on the task of community detection and propose i) a local algorithm that reveals the community structure of a vertex and easily facilitates distributed execution and ii) a streaming algorithm that greatly outperforms non-streaming state-of-the-art approaches with respect to both execution time and memory usage. In addition, we propose a streaming sampling technique that allows for capturing the interesting part of an unmanageable volume of data produced by social activity. Finally, we exploit the available data of a popular social networking site to empirically investigate a well-studied opinion formation model, using a distributed algorithm.

Keywords: Distributed graph processing · streaming graphs · graph compression · community detection · opinion formation.

1 Introduction

Real-life systems involving interacting objects are typically modeled as graphs and can often grow very large in size. A multitude of contemporary applications heavily involves such graph data and has driven to research directions that allow

* Dissertation Advisor: Alex Delis, Professor

for efficient handling of large scale networks. Two prominent such directions are distributed graph processing and streaming graph algorithms.

The tremendous growth of the Web graph has driven Google to introduce Pregel, a scalable platform with an API that allows for expressing arbitrary graph algorithms. Pregel is a distributed graph processing system that powers the computation of PageRank and has served as an inspiration to many systems that adopted its programming model. One such system is Apache Giraph which originated as the open-source counterpart of Pregel. Giraph is maintained by developers of Facebook that use it to analyze Facebook's social graph. Pregel-like systems follow a vertex-centric approach and address the task of in-memory batch processing of large scale graphs [26]. Communication details are abstracted away from the developers that implements algorithms for such systems. The latter offer APIs that allow for specifying computations with regard to what each vertex of the graph needs to compute whereas edges serve the purpose of transmitting results from one vertex to another. The input graph is loaded on start-up and the entire execution takes place in-memory. Consequently, the execution of a graph algorithm in a Pregel-like system depends on the available memory and will fail if the later is not sufficient enough to fit the graph.

The ever-increasing size of real-world networks has also motivated the design of algorithms that process massive graphs in the data stream model [42]. More specifically, the input of algorithms in this model is defined by a stream of data which usually comprises the edges of the graph. Therefore, graph stream algorithms are a perfect fit for problems dealing with networks that are formed as we attempt to analyze them, e.g., the network describing the activity taking place in a social networking site. However, many challenges arise in a streaming setting that need to be addressed when designing respective techniques. A streaming graph algorithm processes the stream in the order it arrives and each element of the stream must be processed immediately or stored as it will not become available again. In addition, the size of the stream and the speed in which its elements arrive do not allow for persisting the stream in its entirety. Therefore, processing cannot occur at a later stage.

In this dissertation we focused on both distributed and streaming graph processing techniques. We initially investigated the memory usage patterns that contemporary distributed graph processing systems adopt. We observed that graph compression techniques have not been considered in the design of the representations that distributed systems employ. Therefore, we built on compression techniques that assume centralized execution and provided numerous novel compact representations that are fitting for all Pregel-like systems. Our structures offer memory-optimization regardless of the algorithm that is to be executed, and enable the successful execution of algorithms in settings that state-of-the-art systems fail to terminate. We continued by studying a problem that has received considerable attention in the past, yet is still extremely relevant as previously proposed approaches fail to handle the massive volume of today's real-world graphs. In particular, we addressed the problem of community detection and our contribution was twofold as we proposed both a vertex-centric and

a streaming approach. We followed the trend of seed-set expansion methods in which small sets of nodes are expanded to communities. Our techniques offer impressive results with regards to all accuracy, memory usage and execution time. Next, we considered the stream of real-time activity of a social networking site and investigated ways of deriving the interesting part out of it based on network properties. More specifically, we used the interactions of the site's users to construct a network of authorities and assess whether each particular element of activity in the stream is interesting. This approach enables applications in numerous fields to exploit in real-time the enormous amount of information that is made available online everyday without being overwhelmed by the volume of the information. Finally, we investigated yet another field of study in the area of graph mining, namely opinion formation. We adopted a well-studied model and employed a distributed graph processing system to evaluate whether the predicted behavior of the users of a real social network according to this model matches the actual behavior these users.

Most of these results have appeared in [33–38]. What follows is a brief presentation of the topics and results of the dissertation, avoiding technical details.

2 Memory-optimized Distributed Graph Processing

The proliferation of web applications, the explosive growth of social networks, and the continually-expanding WWW-space have led to systems that routinely handle voluminous data modeled as graphs. *Facebook* has over 1 billion active users [15] and *Google* has long reported that it has indexed unique URLs whose number exceeds 1 trillion [2]. This ever-increasing requirement in terms of graph-vertices has led to the realization of a number of distributed graph-processing approaches and systems [1, 45, 40]. Their key objective is to efficiently handle large-scale graphs using predominantly commodity hardware [26]. Most of these approaches parallelize the execution of algorithms by dividing graphs into partitions [47] and assigning vertices to workers following the "*think like a vertex*" programming paradigm introduced with `Pregel` [41]. However, recent studies [26, 13] point out that the so-far proposed frameworks [1, 45, 40] fail to handle the unprecedented scale of real-world graphs as a result of ineffective, if not right out poor, memory usage [26]. Thereby, the space requirements of real-world graphs have become a major memory bottleneck.

Deploying space-efficient graph representations in a vertex-centric distributed environment to attain memory optimization is critical when dealing with web-scale graphs and remains a challenge. Related efforts have exclusively focused on providing a compact representation of a graph in a centralized machine environment [9, 5, 14, 39]. In such single-machine settings, we can exploit the fact that vertices tend to exhibit similarities. However, this is infeasible when graphs are partitioned on a vertex basis, as each vertex must be processed independently of other vertices. Furthermore, to achieve memory optimization, we need representations that allow for mining of the graph's elements *without decompres-*

sion; this decompression would unfortunately necessitate additional memory to accommodate the resulting unencoded representation.

A noteworthy step towards memory optimization was taken by *Facebook* when it adopted `Apache Giraph` [1] for its graph search service; the move yielded both improved performance and scalability [15]. However, *Facebook*'s improvements regarding memory optimization entirely focused on a more careful implementation for the representation of the out-edges of a vertex [15]; the redundancy due to properties exhibited in real-world graphs was not exploited.

We investigate approaches that help realize compact representations of out-edges in (weighted) graphs of web-scale while following the `Pregel` paradigm. The vertex placement policy that `Pregel`-like systems follow necessitates for storing the out-edges of each vertex independently. This policy preserves the *locality of reference* property, known to be exhibited in real-world graphs [8], and enables us to exploit in this work, patterns that arise among the out-edges of a *single* vertex. We cannot however utilize similarities among out-edges of different vertices, for we are unaware of the partition each vertex is placed into. Our first technique, termed `BVEdges`, applies all methods proposed in [9] that can effectively function with the vertex placement policy of `Pregel` in a distributed environment. `BVEdges` primarily focuses on identifying intervals of consecutive out-edges of a vertex and employs universal codings to efficiently represent them. To facilitate access without imposing the significant computing overheads of `BVEdges`, we propose `IntervalResidualEdges`, which holds the corresponding values of intervals in a non-encoded format. We facilitate support of weighted graphs with the use of a parallel array holding variable-byte encoded weights, termed `VariableByteArrayWeights`. Additionally, we propose `IndexedBitArrayEdges`, a novel technique that considers the out-edges of each vertex as a single row in the adjacency matrix of the graph and indexes only the areas holding edges using byte sized bit-arrays. Finally, we propose a fourth space-efficient tree-based data structure termed `RedBlackTreeEdges`, suitable for algorithms requiring mutations of out-edges.

Our experimental results with diverse datasets indicate significant improvements on space-efficiency for all our proposed techniques. We reduce memory requirements up–to 5 times in comparison with currently applied methods. This eases the task of scaling to *billions of vertices per machine* and so, it allows us to load much larger graphs than what has been feasible thus far. In settings where earlier approaches were also capable of executing graph algorithms, we achieve significant performance improvements in terms of time of up–to 41%. We attribute this to our introduced memory optimization as less time is spent for garbage collection. These findings establish our structures as the undisputed preferable option for web graphs, which offer compression-friendly orderings, or any other type of graph after the application of a reordering that favors its compressibility. Last but not least, we attain a significantly improved trade-off between space-efficiency and performance of algorithms requiring mutations through a representation that uses a tree structure and does not depend on node orderings.

3 Uncovering Local Hierarchical Link Communities at Scale

The neurons in our brains, the proteins in live cells, the powerplants of an electrical grid, and the users of an online social networking service, are all entities of *complex systems* that play a vital role in our daily lives. Networks are a powerful tool for modeling relations and interactions between the components of such complex systems. Respective real-world networks are often massive; yet they exhibit a high level of order and organization, which allows the study of common properties they exhibit, such as the power-law degree distribution and the small-world structure [46, 19]. Another important property that real-world networks exhibit is the presence of community structure [24]. At a high level, communities are groups of nodes that share a common functional property or context, e.g., two people that attended the same school, or two movies with the same actor. In several cases communities in a network are distinct; consider for example the fans of different basketball teams. However, it is often the case that communities overlap.

Effectively extracting the community structure of a node in a network has many useful applications, e.g., i) we can provide more informative and engaging social network feeds by better understanding the membership of an individual to various organizational groups, and ii) we can suggest common friends of an individual to connect because they share mutual interests. Early community detection approaches focused either on grouping the nodes of a network or on searching for links that should be removed to separate the clusters [20]. However, these approaches did not consider the fact that communities may overlap, and ultimately could not provide an accurate representation of a network's community structure. Algorithms that followed [4, 25, 48] allow for nodes to belong to several overlapping communities by employing techniques such as link clustering, matrix factorization, and personalized PageRank vectors. Still, these approaches are not applicable to the massive graphs of the *Big Data* era, as they focus on the *entire* graph structure and do not scale with regards to both execution time and memory consumption. Recent efforts have therefore shifted the focus from the global structure to a local view of the network [30–32]. More specifically, such approaches locally expand a set of target nodes in the community of interest, instead of uncovering the communities of the entire network.

Seed set expansion approaches employ techniques such as random walks to estimate the likelihood of a node to participate in the target community, and manage to scale to large networks [30–32]. These approaches consider that overlaps between communities are sparsely connected whereas the areas where communities overlap are denser than the actual communities. However, studies of real-world networks show that two nodes are more likely to be connected if they share multiple communities in common [49]. Hence, as the overlapping area is in fact denser than the actual communities, seed set expansion approaches are driven towards nodes that reside in the overlap. In addition to this, all scalable methods require *multiple seeds* to avoid detecting multiple overlapping communities as a single one. This constitutes a challenge, as it is usually the case that

we are interested in all communities of a single node, instead of seeking one community involving multiple predefined nodes. Finally, seed set expansion approaches are shown to perform well when detecting relatively large communities, whereas high quality communities are in fact small [49].

Here, we focus on the neighbors of a single node in the network, i.e., its egonet, and aim at extracting the –possibly overlapping– communities of this node. We build upon the ideas of *link clustering* [4, 18] and employ *similarity* measures that allow for effectively handling densely connected overlaps between communities. Our intuition is that when grouping pairs of links we should capture the *extent* to which a link belongs to multiple overlapping communities. To this end, we utilize a dispersion-based tie-strength measure that helps us quantify the participation of a link's adjacent nodes to more than one community. Our approach is both *efficient* and *scalable* as we focus on local parts of graphs comprising a target node and its neighbors. As we show through experimental evaluation, we produce a more accurate and intuitive representation of the community structure around a node for a number of real-world networks.

4 Community Detection via Seed Set Expansion on Graph Streams

Graph structures attract significant attention as they allow for representing entities of various domains as well as the relationships these entities entail. Real-world networks are commonly portrayed using graphs and are often massive. Despite their size, such networks exhibit a high level of order and organization, a property frequently referred to as community structure [24]. Nodes tend to organize into densely connected groups that exhibit weak ties with the rest of the graph. We refer to such groups as communities, whereas the task of identifying them is termed *community detection*.

Community detection is a fundamental problem in the study of networks and becomes more relevant with the prevalence of online social networking services such as Twitter and Facebook. Identifying the social communities of an individual enables us to perform recommendations for new connections. Moreover, by better understanding the membership of an individual to various organizational groups, we can provide more informative and engaging social network feeds. In addition to social networks, community detection is successfully applied to numerous other types of networks, such as biological or citation networks. In the former, we are particularly interested in inferring communities of interacting proteins, whereas in the latter we wish to uncover relationships between disciplines or the citation patterns of authors [20].

In the last two decades a plethora of community detection methods has been proposed [7, 16, 43, 44, 4, 25, 48]. However, these approaches are not applicable to the massive graphs of the Big Data era, as they focus on the *entire* graph structure and do not scale with regards to both execution time and memory consumption. Recent efforts manage to scale as far as execution time is concerned by focusing on the local structure and expanding exemplary seeds-sets

into communities [30, 32, 33]. Such a seed-set expansion setting can be applied to numerous real world applications, e.g., given a few researchers focusing on Big Data we can use a citation network to detect their colleagues in the same field. However, the space requirements of such algorithms rapidly become a concern due to the unprecedented size now reached by real-world graphs. The latter have become difficult to represent in-memory even in a distributed setting [37].

An increasingly popular approach for massive graph processing is to consider a *data stream model*, in which the stream comprises the edges of a graph [42]. This is a new direction in the field of community detection and to the best of our knowledge no prior approach has considered such a setting without imposing restrictions on the order in which edges are made available [27, 50]. We propose CoEuS, a novel community detection algorithm that is fully applicable on graph streams. CoEuS is initialized with seed-sets of nodes that define different communities. As edges arrive, we can process them but we cannot afford to keep them all in-memory. Therefore, CoEuS maintains rather limited information about the adjacent nodes of each edge and their participation in the communities in question. This information is kept using probabilistic data structures to further reduce the memory requirements of our algorithm. In addition to our original idea for community detection in graph streams, we propose two algorithms to enhance the effectiveness of CoEuS. The first one focuses on better quantifying the quality of each edge w.r.t. to a community. The second one is a novel clustering algorithm that allows for automatically determining the size of the resulting communities, in spite of the absence of the graph structure.

Our experimental results on various large scale real-world graphs show that CoEuS is extremely competitive with regard to *accuracy* against approaches that employ the entire graph structure and cannot operate on graph streams. More specifically, CoEuS can process with just a few MBs, graphs that prior approaches fail to handle on a machine with 16GB of RAM. Moreover, CoEuS is able to derive the communities in question inordinately faster. More importantly, CoEuS is able to return its resulting communities *on demand* at any time as we process the graph stream. This is particularly important, as even if we could afford to use space linear to the number of a graph's edges, no other approach is able to update communities as new edges arrive with no additional *significant* computational cost.

5 Adaptively Sampling Authoritative Content from Social Activity Streams

The tremendous scale of content generation in online social networks brings several challenges to applications such as content recommendation, opinion mining, sentiment analysis, or emerging news detection, all of which have an inherent need to mine this content in real time. As an example, the daily volume of new *tweets* posted by users of `Twitter` surpasses 500 million.[1] However, not

[1] http://www.internetlivestats.com/twitter-statistics/

all generated online social activity is useful or interesting to all applications. Using `Twitter` again as an example, more than 90% of its posts is actually conversational and of interest strictly limited to a handful of users, or spam [23]. Therefore, applications such as emerging news detection that operate on the entire stream, spend a lot of computational cycles as well as storage in processing posts that are not very useful.

One way to solve this problem is, instead of processing the social activity stream in its entirety, to take a sample of the activity and operate on the sample. Through sampling, our goal is to still capture the important and interesting parts of the activity stream, while reducing the amount of data that we would have to process. To this end, one obvious approach is to perform random sampling, i.e., randomly pick a subset of the activity stream and use that in the respective application. A more effective approach however, is to sample content published in the activity stream only from the users that are considered authoritative (or *authorities*).[2] By sampling the posts of authoritative users from the stream, we are reportedly [51] more likely to produce samples that are of *high-quality*, with limited conversational content and less spam.

The challenge in sampling high quality content from a social activity stream lies therefore in identifying authoritative users. Existing work deploys white-lists of users that are likely to produce authoritative content [22, 51] and samples their activity. Although such approaches have been shown to work well for certain applications, we will show experimentally that they are unable to cope with the dynamic nature of a social activity stream where, for example, new users emerge as authorities and old ones fade out. Other prior efforts on identifying authoritative users in social networks (not streams) have focused on computing a relative ranking of users based on network attributes [3, 11, 12, 28, 52]. We build on the findings of such approaches to identify authorities likely to produce useful content; our approach is different however, as we cannot presume that the complete structure of the social network is available, nor that we can afford to process the network offline.

We operate with the more practical assumption that we have incomplete access to the social network. In other words, we do not know which users exist in the network but we simply observe some partial activity from a social activity stream. Our goal is to produce high quality samples from such streams that will still be as useful as possible compared to being able to access the entirety of the social network and the activity within.

We propose RHEA,[3] an adaptive algorithm for sampling authoritative social activity content. RHEA forms a *network of authorities* as it processes a stream and includes in its sample only the content published by the top-K authorities in this network. Given a social activity stream with user interactions (e.g., answers in `Q&A` sites or mentions in the case of `Twitter`) we create a weighted graph used to quantify user authoritativeness. To deal with the potentially enormous

[2] We use terms *authoritative users* and *authorities* interchangeably.

[3] Rhea was the Titaness daughter of the earth goddess Gaia and the sky god Uranus. Her name stands for "she who flows".

amount of items that we encounter in the stream and limit memory blowup, we construct a highly compact, yet extremely efficient sketch-based novel data structure to maintain the authoritative users of the network. Our experimental results with half a billion posts from two popular social networks show significant improvements with regard to various binary and ranked retrieval measures over previous approaches. RHEA is able to sample significantly more *relevant* documents, with *higher precision* and remarkably more accurate *ranking* compared to sampling based on static white-lists of authoritative users. Our approach is generic and can be used with any online social activity stream, as long as we can observe indicators of authoritativeness in the stream.

6 On the Impact of Social Cost on Opinion Dynamics

An ever-increasing amount of social activity information is available today, due to the exponential growth of online social networks. The structure of a network and the way the interaction among its users impacts their behavior has received significant interest in the sociology literature for many years. The availability of such rich data now enables us to analyze user behavior and interpret sociological phenomena at a large scale. *Social influence* is one of the ways in which social ties may affect the actions of an individual, and understanding its role in the spread of information and opinion formation is a new and interesting research direction that is extremely important in social network analysis. The existence of social influence has been reported in psychological studies [29] as well as in the context of online social networks [10]. The latter usually allow users to endorse articles, photos or other items, thus expressing shortly their opinion about them. Each user has an internal opinion, but since she receives a feed informing her about her friends' endorsements, her expressed (or overall) opinion may well be influenced by her friends' opinions. This process may lead to a consensus.

The most notable example of studying consensus formation due to information transmission is the DeGroot model [17]. This model considers a network of individuals with an opinion which they update using the average opinion of their friends, eventually reaching a shared opinion. In [21] the notion of an individual's internal opinion is added, which, unlike her expressed opinion, is not altered due to social interaction. This model captures more accurately the fact that consensus is rarely reached in real word scenarios. The popularity of a specific article, for instance, may vary largely between different communities in a social network. This fact gives rise to the study of the lack of consensus, and the quantification of the social cost that is associated with disagreement [6]; the authors here consider a game where the utilities are the users' social costs and perform repeated averaging to get the Nash equilibrium. The resulting models of opinion dynamics in which consensus is not in general reached allow for testing against real-world datasets, and enable the verification of influence existence. Investigating game theoretic models of networks against real data is crucial in understanding whether the behavior they portray depicts an illustration that is close to the real picture.

We study the spreading of opinions in social networks, using a variation of the DeGroot model [21] and the corresponding game detailed in [6]. We perform an extensive analysis on a large sample of a popular social network and highlight its properties to indicate its appropriateness for the study of influence. The observations we make verify our intuitions regarding the source and presence of social influence. Furthermore, we initialize instances of games using real data and use repeated averaging to calculate their Nash equilibrium. We experimentally show that our model, when properly initialized, is able to mimic the original behavior of users and captures the social cost affecting their activity more accurately than a classification model utilizing the same information.

7 Conclusions

In this dissertation we studied two research directions that allow for handling large-scale graphs, i.e., *distributed graph processing* and *streaming graph algorithms*. Our focus was on improving contemporary distributed systems, introducing novel techniques for important graph processing problems, and employing scalable platforms to empirically study real-world networks. We proposed techniques to efficiently address challenges regarding: i) memory-optimized distributed graph processing, ii) large scale distributed and streaming community detection, iii) sampling authoritative content from streams of social activity, and iv) modeling the behavior of social network users. Our contribution in all above areas through extensive experimentation is shown to be significant.

References

1. Apache Giraph. http://giraph.apache.org/
2. We knew the web was big.... http://googleblog.blogspot.ca/2008/07/we-knew-web-was-big.html
3. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding high-quality content in social media. In: WSDM 2008. pp. 183–194
4. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature **466**(7307), 761–764 (2010)
5. Apostolico, A., Drovandi, G.: Graph compression by BFS. Algorithms **2**(3), 1031–1044 (2009)
6. Bindel, D., Kleinberg, J.M., Oren, S.: How bad is forming your own opinion? In: FOCS. pp. 57–66 (2011)
7. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), P10008 (2008)
8. Boldi, P., Rosa, M., Santini, M., Vigna, S.: Layered label propagation: a multiresolution coordinate-free ordering for compressing social networks. In: Proc. of the 20th Int. Conf. on World Wide Web. pp. 587–596 (2011)
9. Boldi, P., Vigna, S.: The webgraph framework I: compression techniques. In: Proc. of the 13th Int. Conf. on World Wide Web, May 17-20. pp. 595–602 (2004)

10. Bond, R.M., Fariss, C.J., Jones, J.J., Kramer, A.D., Marlow, C., Settle, J.E., Fowler, J.H.: A 61-million-person experiment in social influence and political mobilization. Nature **489**(7415), 295–298 (2012)
11. Bouguessa, M., Romdhane, L.B.: Identifying authorities in online communities. ACM Trans. Intell. Syst. Technol. **6**(3), 30:1–30:23 (2015)
12. Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., Vesci, G.: Choosing the right crowd: expert finding in social networks. In: EDBT '13. pp. 637–648
13. Cai, Z., Gao, Z.J., Luo, S., Perez, L.L., Vagena, Z., Jermaine, C.M.: A comparison of platforms for implementing and running very large scale machine learning algorithms. In: SIGMOD 2014, June 22-27. pp. 1371–1382 (2014)
14. Chierichetti, F., Kumar, R., Lattanzi, S., Mitzenmacher, M., Panconesi, A., Raghavan, P.: On compressing social networks. In: KDD 2009, June 28 - July 1. pp. 219–228 (2009)
15. Ching, A., Edunov, S., Kabiljo, M., Logothetis, D., Muthukrishnan, S.: One Trillion Edges: Graph Processing at Facebook-Scale. Proc. of the VLDB Endowment **8**(12), 1804–1815 (2015)
16. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Physical review E **70**(6), 066111 (2004)
17. DeGroot, M.H.: Reaching a consensus. Journal of the ASA **69**(345), 118–121 (1974)
18. Evans, T., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. Physical Review E **80**, 016105 (2009)
19. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: ACM SIGCOMM computer communication review. vol. 29, pp. 251–262. ACM (1999)
20. Fortunato, S.: Community detection in graphs. Physics Reports **486**(3), 75–174 (2010)
21. Friedkin, N.E., Johnsen, E.C.: Social influence and opinions. Journal of Mathematical Sociology **15**(3-4), 193–206 (1990)
22. Ghosh, S., Sharma, N.K., Benevenuto, F., Ganguly, N., Gummadi, P.K.: Cognos: crowdsourcing search for topic experts in microblogs. In: SIGIR '12. pp. 575–590
23. Ghosh, S., Zafar, M.B., Bhattacharya, P., Sharma, N.K., Ganguly, N., Gummadi, P.K.: On sampling the wisdom of crowds: random vs. expert sampling of the twitter stream. In: CIKM'13. pp. 1739–1744
24. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proc. of the National Academy of Sciences **99**(12), 7821–7826 (2002)
25. Gleich, D.F., Seshadhri, C.: Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In: KDD 2012. pp. 597–605 (2012)
26. Han, M., Daudjee, K., Ammar, K., Özsu, M.T., Wang, X., Jin, T.: An Experimental Comparison of Pregel-like Graph Processing Systems. Proc. of the VLDB Endowment **7**(12), 1047–1058 (2014)
27. Hollocou, A., Maudet, J., Bonald, T., Lelarge, M.: A linear streaming algorithm for community detection in very large networks. ArXiv e-prints (Mar 2017)
28. Jurczyk, P., Agichtein, E.: Discovering authorities in question answer communities by using link analysis. In: CIKM 2007. pp. 919–922
29. Kelman, H.C.: Compliance, identification, and internalization: Three processes of attitude change. Journal of conflict resolution pp. 51–60 (1958)
30. Kloster, K., Gleich, D.F.: Heat kernel based community detection. In: KDD '14. pp. 1386–1395 (2014)
31. Kloumann, I.M., Kleinberg, J.M.: Community membership identification from small seed sets. In: KDD '14, August 24 - 27. pp. 1366–1375 (2014)

32. Li, Y., He, K., Bindel, D., Hopcroft, J.E.: Uncovering the small community structure in large networks: A local spectral approach. In: WWW 2015 (2015)
33. Liakos, P., Ntoulas, A., Delis, A.: Scalable link community detection: A local dispersion-aware approach. In: 2016 IEEE International Conference on Big Data, BigData 2016, Washington DC, USA, December 5-8, 2016. pp. 716–725 (2016)
34. Liakos, P., Ntoulas, A., Delis, A.: COEUS: community detection via seed-set expansion on graph streams. In: 2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017. pp. 676–685 (2017)
35. Liakos, P., Ntoulas, A., Delis, A.: Rhea: Adaptively sampling authoritative content from social activity streams. In: 2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017. pp. 686–695 (2017)
36. Liakos, P., Papakonstantinopoulou, K.: On the impact of social cost in opinion dynamics. In: Proceedings of the 10th Int. Conf. on Web and Social Media, Cologne, Germany, May 17-20, 2016. pp. 631–634 (2016)
37. Liakos, P., Papakonstantinopoulou, K., Delis, A.: Memory-optimized distributed graph processing through novel compression techniques. In: Proc. of the 25th ACM Int. Conf, on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016. pp. 2317–2322
38. Liakos, P., Papakonstantinopoulou, K., Delis, A.: Realizing memory-optimized distributed graph processing. IEEE Trans. Knowl. Data Eng. **30**(4), 743–756 (2018)
39. Liakos, P., Papakonstantinopoulou, K., Sioutis, M.: Pushing the Envelope in Graph Compression. In: Proc. of the 23rd ACM Int. Conf. on Information and Knowledge Management. pp. 1549–1558. Shanghai, China (2014)
40. Low, Y., Gonzalez, J., Kyrola, A., Bickson, D., Guestrin, C., Hellerstein, J.M.: Distributed GraphLab: A Framework for Machine Learning in the Cloud. Proc. of the VLDB Endowment **5**(8), 716–727 (2012)
41. Malewicz, G., Austern, M.H., Bik, A.J.C., Dehnert, J.C., Horn, I., Leiser, N., Czajkowski, G.: Pregel: A System for Large-Scale Graph Processing. In: SIGMOD 2010, June 6-10. pp. 135–146 (2010)
42. McGregor, A.: Graph stream algorithms: a survey. SIGMOD Record **43**(1), 9–20
43. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (Feb 2004)
44. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: Computer and Information Sciences-ISCIS 2005, pp. 284–293 (2005)
45. Salihoglu, S., Widom, J.: GPS: a graph processing system. In: SSDBM 2013, July 29 - 31. pp. 22:1–22:12 (2013)
46. de Sola Pool, I., Kochen, M.: Contacts and influence. Social networks **1**(1), 5–51 (1978)
47. Ugander, J., Backstrom, L.: Balanced Label Propagation for Partitioning Massive Graphs. In: WSDM 2013, February 4-8. pp. 507–516 (2013)
48. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: WSDM 2013. pp. 587–596 (2013)
49. Yang, J., Leskovec, J.: Structure and overlaps of ground-truth communities in networks. ACM Trans. on Intelligent Systems and Technology **5**(2), 26 (2014)
50. Yun, S., Lelarge, M., Proutière, A.: Streaming, memory limited algorithms for community detection. In: NIPS 2014, December 8-13. pp. 3167–3175 (2014)
51. Zafar, M.B., Bhattacharya, P., Ganguly, N., Ghosh, S., Gummadi, K.P.: On the wisdom of experts vs. crowds: Discovering trustworthy topical news in microblogs. In: CSCW 2016. pp. 437–450
52. Zhang, J., Ackerman, M.S., Adamic, L.A.: Expertise networks in online communities: structure and algorithms. In: WWW 2007. pp. 221–230