**Computing top-***k* **Closeness Centrality Faster in Unweighted Graphs**

ELISABETTA BERGAMINI, Karlsruhe Institute of Technology MICHELE BORASSI, IMT Institute for Advanced Studies Lucca

PIERLUIGI CRESCENZI, Università degli Studi di Firenze and Université de Paris ANDREA MARINO, Università degli Studi di Firenze

HENNING MEYERHENKE, Karlsruhe Institute of Technology and Humboldt-Universität zu Berlin

Given a connected graph *G* = (*V* , *E*), where *V* denotes the set of nodes and *E* the set of edges of the graph, the length (that is, the number of edges) of the shortest path between two nodes *v* and *w* is denoted by *d* (*v*, *w* ).

1

The closeness centrality of a vertex *v* is then defined as . *n*−*d* ,*w* ) , where *n* = |*V* |. This measure is widely

*w* ∈*V*

(*v*

used in the analysis of real-world complex networks, and the problem of selecting the *k* most central vertices has been deeply analyzed in the last decade. However, this problem is computationally not easy, especially for large networks: in the first part of the article, we prove that it is not solvable in time ( *E* 2−*ϵ* ) on directed graphs, for any constant *ϵ* > 0, under reasonable complexity assumptions. Furthermore, we propose a new algorithm for selecting the *k* most central nodes in a graph: we experimentally show that this algorithm improves significantly both the textbook algorithm, which is based on computing the distance between all pairs of vertices, and the state of the art. For example, we are able to compute the top *k* nodes in few dozens of seconds in real-world networks with millions of nodes and edges. Finally, as a case study, we compute the 10 most central actors in the Internet Movie Database (IMDB) collaboration network, where two actors are linked if they played together in a movie, and in the Wikipedia citation network, which contains a directed edge from a page *p* to a page *q* if *p* contains a link to *q*.

O | |

CCS Concepts: • **Human-centered computing** → **Social network analysis**; • **Mathematics of computing** → **Graph algorithms**;

Additional Key Words and Phrases: Centrality, closeness, complex networks

E. Bergamini’s and H. Meyerhenke’s work is partially supported by German Research Foundation (DFG) grant ME 3619/3-2 within the Priority Programme 1736 *Algorithms for Big Data*. P. Crescenzi’s and A. Marino’s work is partially supported by the Italian Ministry of Education, University, and Research (MIUR) under PRIN 2012C4E3KT national research project AMANDA—Algorithmics for MAssive and Networked DAta. This paper is an extended and improved version of the con- ference paper (Bergamini et al. 2016). A preliminary study appeared in (Borassi et al. 2015b). Authors’ addresses: E. Bergamini, Institute of Theoretical Informatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany; M. Borassi, IMT Institute for Advanced Studies Lucca, Italy; P. Crescenzi, Université de Paris, IRIF, Bâtiment Sophie Germain, 8 Place Aurélie Nemours, 75205 Paris Cedex 13, France; email: pierluigi.crescenzi@irif.fr; A. Marino, Dipartimento di Statistica, Informatica, Applicazioni ‘G. Parenti’ (DiSIA), Università degli Studi di Firenze, Viale Mor- gagni 59, 50134 Firenze, Italy; email: andrea.marino@unifi.it; H. Meyerhenke, Department of Computer Science, Humboldt- Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany; email: [meyerhenke@hu-berlin.de.](mailto:meyerhenke@hu-berlin.de)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2019 Association for Computing Machinery. 1556-4681/2019/09-ART53 $15.00

**53**

<https://doi.org/10.1145/3344719> ACM Transactions on Knowledge Discovery from Data, Vol. 13, No. 5, Article 53. Publication date: September 2019.

53:2 E. Bergamini et al.

**ACM Reference format:**

Elisabetta Bergamini, Michele Borassi, Pierluigi Crescenzi, Andrea Marino, and Henning Meyerhenke. 2019. Computing top-*k* Closeness Centrality Faster in Unweighted Graphs. *ACM Trans. Knowl. Discov. Data* 13, 5, Article 53 (September 2019), 40 pages.

<https://doi.org/10.1145/3344719>

# INTRODUCTION

The problem of identifying the most central nodes in a network is a fundamental question that has been asked many times in a plethora of research areas, such as biology, computer science, sociology, and psychology. Because of the importance of this question, dozens of centrality measures have been introduced in the literature (for a recent survey, see (Boldi and Vigna 2014)). Among these measures, closeness centrality is certainly one of the oldest and of the most widely used (Bavelas 1950): almost all books dealing with network analysis discuss it (for example, (Newman 2010)), and almost all existing network analysis libraries implement algorithms to compute it.

Given a connected graph *G* = (*V* , *E*), where *V* denotes the set of nodes and *E* the set of edges of

the graph, the length (that is, the number of edges) of the shortest path between two nodes *v* and *w*

is denoted by *d* (*v*, *w* ). The closeness centrality of a vertex *v* is then defined as *c* (*v*) = .

*n*−*d*1 ,*w* ) ,

where *n* = | |

*w* ∈*V* (*v*

*V* . The idea behind this definition is that a central node should be very efficient in spreading information to all other nodes: for this reason, a node is central if the average number of links needed to reach another node is small. If the graph is not (strongly) connected, researchers have proposed various ways to extend this definition: for the sake of simplicity, we focus on Lin’s index, because it coincides with closeness centrality in the connected case and because it is quite established in the literature (Lin 1976; Wasserman and Faust 1994; Boldi and Vigna 2013, 2014; Olsen et al. 2014). However, our algorithms can be adapted very easily to compute other possible generalizations, such as harmonic centrality (Marchiori and Latora 2000) and exponential central- ity (Wang and Tang 2014) (see Section 2 for more details).

In order to compute the *k* vertices with largest closeness, the textbook algorithm computes *c* (*v*) for each *v* and returns the *k* largest found values. The main bottleneck of this approach is the computation of *d* (*v*, *w* ) for each pair of vertices *v* and *w* (that is, solving the All Pairs Shortest Paths or APSP problem). This can be done in two ways: either by using fast matrix multiplication (for which the currently best algorithms require time (*n*2.373) (Williams 2012; Gall 2014)) or by

O

performing a breadth-first search (in short, BFS) from each vertex *v V* , in time (*mn*), where *m* =

∈ O

*E* . Usually, BFS is preferred because the approaches based on fast matrix multiplication present a reduced numerical stability and large constants hidden in the big O notation (Gall 2012) (at least the ones that have been implemented so far). Also, real-world networks are usually sparse, that is, *m* is not much bigger than *n*. However, even the BFS-based approach is too time-consuming if the input graph is very big (with millions of nodes and hundreds of millions of edges).

| |

Our first result proves that, in the worst case, the BFS-based approach cannot be improved, under reasonable complexity assumptions. In particular, we show that, assuming the Strong Exponential Time Hypothesis (SETH, (Impagliazzo et al. 2001)), the most central vertex cannot be computed

in (*n*2−*ϵ* ) on sparse graphs, for any *ϵ* > 0. Consequently, for any *ϵ* > 0, there is no algorithm to compute the most central vertex in *O* (*m*2−*ϵ* ) on general graphs, since otherwise such an algorithm would compute the most central vertex in *O* (*n*2−*ϵ* ) on sparse graphs.

O

Knowing that the BFS-based algorithm cannot be improved in the worst case, in the second part

of the article we provide a new exact algorithm that performs much better on real-world networks, making it possible to compute the *k* most central vertices in networks with millions of nodes

ACM Transactions on Knowledge Discovery from Data, Vol. 13, No. 5, Article 53. Publication date: September 2019.

Computing top-*k* Closeness Centrality Faster in Unweighted Graphs 53:3

and hundreds of millions of edges. The new approach combines the BFS-based algorithm with a pruning technique: during the algorithm, we compute and update upper bounds on the closeness of all the nodes, and we exclude a node *v* from the computation as soon as its upper bound is “small enough,” that is, we are sure that *v* does not belong to the top *k* nodes. We propose two different strategies to set the initial bounds, and two different strategies to update the bounds during the computation: this means that our algorithm comes in four different variations. The experimental results show that different variations perform well on different kinds of networks, and the best variation of our algorithm drastically outperforms both a probabilistic approach (Okamoto et al. 2008), and the best exact algorithm available until now (Olsen et al. 2014). We have computed for the first time the 10 most central nodes (with respect to the closeness measure) in networks with millions of nodes and hundreds of millions of edges, and we have done so in very little time. A significant example is the wiki-Talk network, which was also used in (Sariyüce et al. 2013), where the authors propose an algorithm to update closeness centralities after edge additions or deletions. Our performance is about 30,000 times better than the performance of the textbook algorithm: if only the most central node is needed, we can recompute it from scratch more than 150 times faster than the geometric average update time in (Sariyüce et al. 2013). Moreover, our approach is not only very efficient, but it is also very easy to code, making it a very good candidate to be implemented in existing graph libraries. We provide an implementation of it in NetworKit (Staudt et al. 2016) and of one of its variations in Sagemath (Csárdi and Nepusz 2006). We sketch the main ideas of the algorithm in Section 4, and we provide all details in Sections 5–8. We experimentally evaluate the efficiency of the new algorithm in Section 9.

Also, our approach can be easily extended to any centrality measure in the form *c* (*v*) = *w* Ç*v f* (*d* (*v*, *w* )), where *f* is a decreasing function. Apart from Lin’s index, almost all the ap- proaches that try to generalize closeness centrality to disconnected graphs fall under this cate- gory. The most popular among these measures is harmonic centrality (Marchiori and Latora 2000),

.

.

defined as *h*(*v*) = *w* Ç*v d* ( 1 *w* ) . For the sake of completeness, in Section 9, we show that our algo- rithm performs well also for this measure.

*v*,

In the last part of the article (Sections 10 and 11), we consider two case studies: the actor collabo- ration network (1,797,446 vertices, 72,880,156 edges) and the Wikipedia citation network (4,229,697 vertices, 102,165,832 edges). In the actor collaboration network, we analyze the evolution of the 10 most central vertices, considering snapshots taken every 5 years between 1940 and 2014. The computation was performed in little more than 45 minutes. In the Wikipedia case study, we con- sider both the standard citation network, that contains a directed edge (*p*, *q*) if *p* contains a link to *q*, and the reversed network, that contains a directed edge (*p*, *q*) if *q* contains a link to *p*. For most of these graphs, we are able to compute the 10 most central pages in a few minutes, making them available for further analyses.

# Related Work

Closeness is a “traditional” definition of centrality, and consequently it was not “designed with scalability in mind,” as stated in (Kang et al. 2011). Also in (Chen et al. 2012), it is said that closeness centrality can “identify influential nodes,” but it is “incapable to be applied in large-scale networks due to the computational complexity.” The simplest solution considered was to define different measures that might be related to closeness centrality (Kang et al. 2011).

*Hardness results*. A different line of research has tried to develop more efficient algorithms, or lower bounds for the complexity of this problem. In particular, in (Borassi et al. 2015a) it is proved that finding the least closeness-central vertex is not subquadratic-time solvable, unless SETH is false. In the same line, it is proved in (Abboud et al. 2016) that finding the most central vertex

ACM Transactions on Knowledge Discovery from Data, Vol. 13, No. 5, Article 53. Publication date: September 2019.