PROCEEDINGS A

rspa.royalsocietypublishing.org

Research

Article submitted to journal

Subject Areas:
Algorithmic Information Theory, Graph Theory

Keywords:
Graph algorithms, Wikipedia link network, Relevance ranking, Personalized PageRank

Author for correspondence:
e-mail: cristian.consonni@acm.org

CycleRank, or There and Back Again: Personalized Relevance Scores from Cyclic Paths on Directed Graphs

Cristian Consonni[2,*], David Laniado[2] and Alberto Montresor[1]

[1]DISI, University of Trento, Trento
[2]Eurecat - Centre Tecnològic of Catalunya, Barcelona
*This work was developed while the author was at DISI, University of Trento.

Surfing the links between Wikipedia articles constitutes a valuable way to acquire new knowledge related to a topic by exploring its connections to other pages. In this sense, *Personalized PageRank* is a well-known option to make sense of the graph of links between pages, and identify the most relevant articles with respect to a given one; its performance, however, is hindered by pages with high indegree that function as hubs and obtain high scores regardless of the starting point. In this work, we present *CycleRank*, a novel algorithm based on cyclic paths aimed at finding the most relevant nodes related to a topic. To compare the results of *CycleRank* with those of *Personalized PageRank* and other algorithms derived from it, we perform three experiments based on different ground truths. We find that *CycleRank* aligns better with readers' behavior as it ranks in higher positions the articles corresponding to links that receive more clicks; it tends to identify in higher position related articles highlighted by editors in the "See also" section; and it is more robust to global hubs of the network having high indegree. Finally, we show that computing *CycleRank* is two orders of magnitude faster than computing the other baselines.

THE ROYAL SOCIETY PUBLISHING

1. Introduction

Wikipedia is one of the biggest and most used sources of knowledge on the Web. As of this writing, it is the fifth most visited website in the world [1]. Wikipedia is not only a huge repository and collaborative effort; it is also a giant hypertext in which each article has links to the concepts that are deemed relevant to it by the editors [2]. The vast network emerging from the collaborative process provides a rich representation of the connections between concepts, entities and pieces of content, aimed at encompassing "the sum of all human knowledge" [3].

This huge graph has been leveraged for a variety of purposes including constructing semantic networks [4], natural language processing [5], cross-cultural studies [6,7], complex networks modelling [8], understanding human navigation of information networks [9]. While one cannot assume that a single article completely encapsulates a concept [10], the link network can be useful in defining the context of an article. Previous research in controversy mapping has shown how this network can be leveraged to analyze the dominating definition of a topic, such as *"Geoengineering"* [11], shedding light on its boundary, context and internal structure. Furthermore, each linguistic community in Wikipedia produces a different network, which allows for comparing the emerging definition of a topic across different language editions [12].

The connections between Wikipedia articles are valuable, but they are also very abundant. The English version has more than 160 million links between its 5.7 million articles [13]. How can one find guidance within this wealth of data? In particular, how can we analyze the network around a specific topic, to characterize its definition as emerging from the connections involving and surrounding it?

The contribution of this work is a novel approach to make sense of the Wikipedia link network, capable of answering *queries* like "Which are the concepts that are more relevant concerning a given topic?"

We translate such inquiries into a graph problem. The topic of interest is represented by one article, i.e. a node in the graph called *reference node*. Given a reference node r, we want to assign a score to every other node in the graph that captures its relevance to r, based on the link structure. The final output is a ranking of nodes, such that the more relevant ones are ranked higher.

One established algorithm to answer this question is *Personalized PageRank*: a variant of *PageRank* where the user can specify one or more nodes as queries (seeds) and obtain a score for all the other nodes in the graph that measures relatedness to the seeds. However, we have found that, when applied in the context of Wikipedia, this algorithm does not produce satisfactory results since it usually includes very general articles in top positions.

To overcome these limitations, we have developed a novel algorithm to find the most relevant nodes in the Wikipedia link network related to a topic. The technique, called *CycleRank*, takes advantage of the cycles that exist between the links and produces a ranking of the different articles related to one chosen by a user. In other words, nodes receive a non-zero score if they can both get to and be reached from the reference node. Instead of walking randomly, the surfer of *CycleRank* goes out to a node, but it finally comes back.[1]

Differently from *PageRank*, *CycleRank* accounts for links in both directions, and it can provide results that are more accurate than those produced by the well-known *Personalized PageRank* algorithm. While we formulate the problem and validate the results for the Wikipedia link network, we believe the *CycleRank* algorithm can be applied to any other context where both incoming and outgoing links are important to measure relevance with respect to a given node.

The paper is organized as follows. We first formalize the problem we want to solve in Section 2. We provide insights on why *Personalized PageRank* is not a good choice in Section 3 and we discuss related work in Section 4. We describe *CycleRank* in Section 5 and we evaluate its performance in Section 6. Conclusions are drawn in Section 7.

[1]This surfer eventually comes back home, just as Bilbo Baggins, the protagonist of *"The Hobbit, or There and Back Again"*, after leaving the Shire to go on an adventure.

2. Problem Statement

Given a graph $G = (V, E)$, where V is a finite set containing n nodes (articles) and $E \subseteq V \times V$ is a set containing m directed edges (links between articles), we are seeking to build a *ranking*, i.e. an order relationship between nodes based on their relevance with respect to a reference node r.

In order To achieve this goal, we build a *ranking function* rf_r that assigns a non-negative score to every node $v \in V$:

$$rf_r : V \mapsto [0, +\infty)$$

The ranking $\nu_r = [v_1, v_2, \ldots, v_n]$ is thus given by the total order of scores: if $rf_r(v_i) > rf_r(v_j)$, then node v_i should appear before node v_j (i.e., $i < j$). Note that we assume that there are no ex-aequo in any given ranking; this can by achieved by breaking ties randomly.

3. Background

The *PageRank* algorithm represents an established relevance measure for directed networks [14]; its variant *Personalized PageRank* may be used to measure relevance within a certain context. *PageRank* is a measure based on incoming connections, where connections from relevant nodes are given a higher weight. Intuitively, the *PageRank* of a node represents the probability that, following a random path in the network, one will reach that node. It is computed in an iterative process, as the *PageRank* of a node depends on the *PageRank* scores of the nodes that link to it. There are however efficient algorithms to compute it. The idea behind *PageRank* is that of simulating a stochastic process in which a user follows random paths in a hyperlink graph. At each round, the user either keeps surfing the graph following the link network with probability α, or is teleported to a random page in the graph with probability $1 - \alpha$. The parameter α is called *damping factor* and is generally assumed to be 0.85 [15,16]. During the surfing process, the algorithm assumes equal probability of following any hyperlink included in a page; similarly, when teleported, every other node in graph can be selected with equal probability.

Personalized PageRank is a variant of the original *PageRank* algorithm, where the user provides a set of *seed* nodes. In *Personalized PageRank*, teleporting is not directed to some random node taken from the entire graph, but to one taken from the seed set. In this way, the algorithm models the relevance of nodes around the selected nodes, as the probability of reaching each of them, when following random walks starting from a node in the seed set.

Limitations of PageRank. At first look, *Personalized PageRank* seems to be suitable for our use case, as it can be used to represent a measure of relevance of Wikipedia articles strongly linked (directly or indirectly) to the seed. However, we found unsatisfactory results when applying this algorithm. Very often, pages that are found to be very central in the overall network, such as *"United States"* or *"The New York Times,"* are included in the top results of completely unrelated queries. Such central articles act as hubs in the graph; they have such a strong relevance overall that, even starting from a seed article which is not specially related to them, one is very likely to end up reaching them while exploring the graph.

We argue that this is due to different factors. First, paths of any length can be followed; therefore, in a densely connected graph many paths will tend to converge towards the most relevant nodes. This can be limited only partially by lowering the value of the damping factor.

Second, *PageRank* only accounts for inlinks, not for outlinks. This is reasonable for web search and other contexts where inlinks are a good proxy for relevance, as they represent somehow the value attributed to a node by the other nodes of the graph. In such cases, outlinks have basically no value: it is very easy to add into one's web page many outlinks to other pages. In the context of Wikipedia, instead, links from an article to other articles may be subject to being inserted and accepted by the editors' community as much as incoming links from other articles. So, both outgoing and incoming links can be considered as indicators of relevance. In particular, outlinks to other pages from an article can be a very valuable indicator that these pages are actually related to the topic. For example, if an article contains links to *"Computer Science,"* then we can assume that

its content is related to *"Computer Science;"* on the other hand, we can expect the article *"United States"* to have only a few links to articles related to *"Computer Science,"* as it is not the main subject of the article.

4. Related Work

We discuss here relevant related studies used to establish the foundation of our work.

Identifying related content in Wikipedia. A few previous studies have addressed the problem of recommending relevant Wikipedia articles, starting from a given article, with different approaches including machine learning [17] and Markov chains [18]. Schwarzer et al. [19] used citation-based document similarity measures, and evaluated their effectiveness using as a ground truth links in the "See also" section of an article, and readers' clicks on the links from an article. With respect to their work, the main difference of our approach is that we focus on the problem of finding relevant related nodes on a graph, and we do not use the text of Wikipedia articles. Our approach has not only the advantage of being completely language-independent, but it is applicable to a much broader set of problems.

Link Structure in Wikipedia. The foundation of this paper is based on the idea that inlinks and outlinks in Wikipedia have a comparable role to establish relevance. Kamps and Koolen [20] performed a comparative analysis of the link structure of Wikipedia and a selection of the Web - built from `.gov` websites - and found that traditional information retrieval algorithms such as HITS do not work well on Wikipedia. The root case of this problem, as they observe, is that in Wikipedia inlinks and outlinks are good indicators of relevance, contrasting the general behavior of the web where only the former provide this indication.

PageRank and variations. Boldi et al. [16] studied the behavior of *Personalized PageRank* as a function of the damping factor α. While they acknowledge that a popular choice of α is 0.85 – following the suggestion of the authors of *PageRank* itself [14] – they discuss both the possibility of choosing smaller value of α as well as values close to 1, finding the latter to be a choice with several theoretical and computational shortcomings. Gleich et al. [15] studied the problem of determining the empirical value for α from the visitor logs of a collection of websites, including Wikipedia. They found Wikipedia visitors do not tend to teleport, and estimated the distribution of the values of α for Wikipedia to a beta distribution with maximum at $\alpha = 0.30$. In our experiments, we have considered $\alpha = 0.30$ and $\alpha = 0.85$ as values for the damping parameter when executing *Personalized PageRank*.

We focus on the variations of *PageRank* that use reverse links or take into account both the existence of inlinks and outlinks. In 2010, Chepelianskii [21] introduced the idea of calculating the pagerank score of nodes on the transposed graph – called *CheiRank*– as well as on the original graph and performed a study of the correlation between the two scores on a collaboration network. Later, Zhirov [22] combined *CheiRank* and *PageRank* to produce a single two-dimensional ranking of Wikipedia articles, *2DRank*. This method does not assign a score to each node, but just produces a ranking. It was used together with *PageRank* to rank biographies across different language editions [7].

Cycles in Non-Directed Graphs. Finally, we present related work about loops in undirected graphs. This area of work is interesting because it provides a broader context in which to insert our algorithm and it could be used as a guide to extend our algorithm to undirected graphs. However, we consider this line of work to be very different in scope and purpose from our current work. It has been shown recently that graphs with different structure can be distinguished from one another using a measure defined with non-backtracking cycles, i. e. a closed walk that does not retrace any edge immediately after traversing them [23]. This method is tied to the idea of using the length spectrum of a graph from its Laplacian matrix. Graph spectra are extensively covered in literature [24].

5. The CycleRank Algorithm

We propose a more general approach to the problem, defining a new measure of the relevance with respect to a given node in a directed network, that accounts for both incoming and outgoing links. We call this measure *CycleRank*, as it is based on the idea of circular random walks.

Starting from the observation that, in *PageRank*, random walks easily lead to paths that are not related to the topic under consideration, we only consider walks coming back to the starting point within a maximum of K steps. In this way, we guarantee that we only touch pages that are, at least indirectly, both linked from and linking to the reference article. Furthermore, no damping factor is needed, as we can assume that all walks just start from the reference node and come back.

Intuitively, a node that is linked from the reference article but does not link to it is likely to be a concept that is not related to that subject, even if it is important to its definition. Specularly, a node that links to the reference article but is not linked from it is likely to be related to it, but not relevant. Nodes that are linked both from and to a reference node are the ones that we expect to be relevant. Extending this principle, we want then to be able to quantify the relevance of a node with respect to a given reference node, accounting also for the indirect links, i.e. for the amount of paths that can be found linking it from and to the reference node. We do this by counting the cycles involving the reference node that pass through a given other node. As short distances represent a stronger relationship, shorter cycles should get higher weights.

We define the *CycleRank* score $CR_r(i)$ of a node i with respect to a reference node r as follows:

$$CR_r(i) = \sum_{k=2}^{K} \ell_r^k(i) \cdot \sigma(k) \tag{5.1}$$

where $\ell_r^k(i)$ is the number of cycles of length k that include both node i and r, K is a parameter representing the maximum length considered for cycles, and $\sigma(\cdot)$ is a scoring function giving different weights to cycles of different length.

In this way, given a reference node r, the *CycleRank* score of a node i represents the number of cycles including both r and node i, weighted by the scoring function, which depends on the length of the cycle. The reference node is also considered in this computation, and it gets the maximum score as by definition it is included in all the cycles considered.

The threshold K is a parameter whose value can be specified according to the context. It can be set to infinite, but it will never exceed the number of nodes n. It can be typically set to a much lower value for two main reasons: to reduce the computational load and to avoid potential noise deriving from long cycles that include popular nodes far from the reference node. For reference, in our experiments we chose to apply thresholds $K = 3$ and $K = 4$, which produce good results with a limited computational effort.

The main *CycleRank* algorithm is shown in Algorithm 1. To optimize the score computation, we first filter the graph G through function FILTERGRAPH(G,r,K), removing those nodes that could never appear in cycles including the reference node r with length limited by K. We then compute the score on this network using function COMPUTESCORE().

Algorithm 1 *CycleRank*

Input: G: a directed graph $G = (V,E)$
Input: r: the refence node
Input: K: threshold parameter, $K \in \mathbb{N}^+$
Output: *score*: a vector of CycleRank scores for each $v \in V$

```
function CycleRank(G, r, K)
    r ← FilterGraph(G,r,k)
    score ← ComputeScore(G,r,k)
    return score
end function
```

(a) Preliminary filtering

To reduce the size of the network, Algorithm 2 employs a known efficient algorithm to compute the distance from and to the reference node r, and discards all the nodes whose cumulative distance (back and forth) is smaller than K:

(i) We compute the distance $df[v]$ of each node v from the reference node by performing a breadth-first visit of the graph, early-terminating the visit when we reach distance K (Algorithm ETBFS – Early-Terminated Breadth-First Search, Line 6);
(ii) We discard all nodes for which $df[i] > K - 1$ (Line 7), including those unreachable from r whose distance is $+\infty$. The function REMOVENODES$(G, key{=}cond)$ eliminates all nodes in G that do not satisfy the condition expressed by the boolean expression $cond$;
(iii) We compute the distance $dt[v]$ on the transposed network, i.e. the distance from each node v to the reference node (Line 9);
(iv) We compute the length $df[v] + dt[v]$ of the minimum cycle including v and the reference node and we discard all nodes for which $df[v] + dt[v] > K$ (Line 10);

Algorithm 2 FILTERGRAPH

Input: G: a directed graph $G = (V,E)$
Input: r: the refence node
Input: K: threshold parameter, $K \in \mathbb{N}^+$
Output: r: the reference node in the filtered graph

```
1:  function FilterGraph(G, r, K)
2:      for v ∈ V do
3:          df[v] ← +∞                          ▷ Distance from the reference node r
4:          dt[v] ← +∞                          ▷ Distance from the reference node r
5:      end for
6:      ETBFS(G, r, K, df)                      ▷ Step 1
7:      removeNodes(G, key=(df[v] > K − 1))
8:      r ← remapNodes(G, r)
9:      ETBFS(G^T, r, K, dt)                    ▷ Step 2
10:     removeNodes(G, key=(df[v] + dt[v] > K))
11:     r ← remapNodes(G, r)
12:     return r
13: end function
```

In this way we discard all the nodes that are not reached by any cycle of length lower than K. We remap node indexes at each step (Lines 8 and 11) so we effectively work with smaller networks. It should be noted that, in case of $K \geq n$, only nodes unreachable from r will be removed, as the length of simple cycles is bounded by the number of nodes n. The removed nodes will all receive a score of zero.

(b) Cycle enumeration

We then proceed to enumerate all simple cycles in the reduced graph. Our algorithm is based on Johnson's algorithm [25], limited to the query node r and early-terminated. Algorithm 3 presents the details. Each node in our algorithm is associated with the following values:

- $score[v]$, the *CycleRank* score of node v
- $blocked[v]$, a boolean indicating whether v cannot be further visited when searching for a cycle because we already went through it. The purpose of this vector is to avoid going through the same node more than once, since we are only interested in simple cycles.

Algorithm 3 COMPUTESCORE

Input: G: a directed graph $G = (V,E)$
Input: r: the refence node
Input: K: threshold parameter, $K \in \mathbb{N}^+$
Output: $score$: a vector of CycleRank scores for each $v \in V$

```
1: function COMPUTESCORE(G, r, K)
2:     for v ∈ V do
3:         score[v] ← 0                          ▷ CycleRank score
4:         blocked[v] ← false
5:         B[v] ← LIST()
6:     end for
7:     S ← STACK()
8:     CIRCUIT(G, r, r, K, S)
9:     return score
10: end function
```

- $B[v]$, a list of nodes that can be unblocked when node v is unblocked.

These variables are then considered global in the rest of the algorithms, to avoid long signatures.

Cycle discovery is performed through a recursive backtrack visit (Algorithm 4). In function CIRCUIT(G, r, v, K, S), G is the graph, r is the reference node, v is the current visited node, K is the threshold and S is a stack of nodes that have been visited so far.

We use K to early-terminate the search for a cycle when we arrive at the maximum length: in Line 3, we check that current whether the current size of the stack is smaller than K, in which case we can proceed in exploring the graph; otherwise, the function returns immediately.

The CIRCUIT() function works by recursively visiting the nodes on the graph; when we visit a node v we add it to the stack S and mark it as blocked, then we visit its neighbors by looping over the adjacent nodes $v.adj$. If the neighbor w we are visiting is the target node, r in our case, then we have found a cyclic path: the score is updated by calling function UPDATESCORE() and the unblocking flag *flag* is set to true. Otherwise, we check if w is unblocked, if so it can be visited and we call *circuit* recursively.

After visiting all the neighbors of v, we check if the current node can be unblocked. Unblocking happens when v is part of a path that formed a cycle. The UNBLOCK(G, v) function at Line 17 is the same as the one defined by Johnson [25] and we omit here for reasons of space. If we unblock a node v, we unblock all the parent nodes that could lead to v, stored in $B[v]$. In this way, we are able to explore alternative paths that form a cycle.

(c) Score computation

Function UPDATESCORE$(score, S)$ updates the score of the nodes recorded in a stack S of length k, by adding $\sigma(k)$ to the score of every node $v \in S$. Several scoring functions σ can be used; in general, a scoring function should capture the idea that longer cycles contribute less.

We use an exponentially decaying function: $\sigma_{\exp}(k) = e^{-k}$, where the length of a cycle is denoted by k. We have chosen the denominator to be exponential in the number of nodes; we present some data to support this choice in the Experimental Evaluation Section where we show that the number of cycles increases more than exponentially with cycle length for our dataset. Intuitively, an exponentially-dacaying scoring function limits the possibility that short cycles become neglectable compared to long cycles in the computation of CycleRank, especially for higher values of k. We empirically validated this intuition by executing the evaluation experiments with linear and quadratic functions; we obtained lower quality results, while with exponential function we obtained the best results in all experiments for our setting. Different scoring functions can be considered based on the problem at hand and according to structural

Algorithm 4 CIRCUIT

Input: G: a directed graph $G(V,E)$
Input: v: a node $v \in V$
Input: r: the reference node $r \in V$
Input: K: a positive integer, $K \in \mathbb{N}^+$
Input: S: a stack of nodes
Output: $flag$: a boolean

```
1:  function CIRCUIT(G, v, r, K, S)
2:      flag ← false
3:      if S.size() < K then
4:          S.push(v)
5:          blocked[v] ← true
6:          for each w ∈ v.adj() do
7:              if w = r then
8:                  UPDATESCORE(score, S)
9:                  flag ← true
10:             else if ¬w.blocked then
11:                 if CIRCUIT(G, w, r, K, S) then
12:                     flag ← true
13:                 end if
14:             end if
15:         end for
16:         if flag then
17:             UNBLOCK(G, v)
18:         else
19:             for each w ∈ v.adj() do
20:                 if v ∉ w.B then
21:                     w.B.push_back(v)
22:                 end if
23:             end for
24:         end if
25:         S.pop()
26:     end if
27:     return flag
28: end function
```

properties of the network; in networks with lower density and clustering coefficient, having a lower amount of cycles, a less skewed σ function may be more suitable.

Algorithm 5 UPDATESCORE

Input: $score$: a stack representing a cycle
Input: S: a stack representing a cycle

```
1: function UPDATESCORE(score, S)
2:     k ← LEN(S)
3:     for each v ∈ S do
4:         score[v] = score[v] + σ(k)
5:     end for
6: end function
```

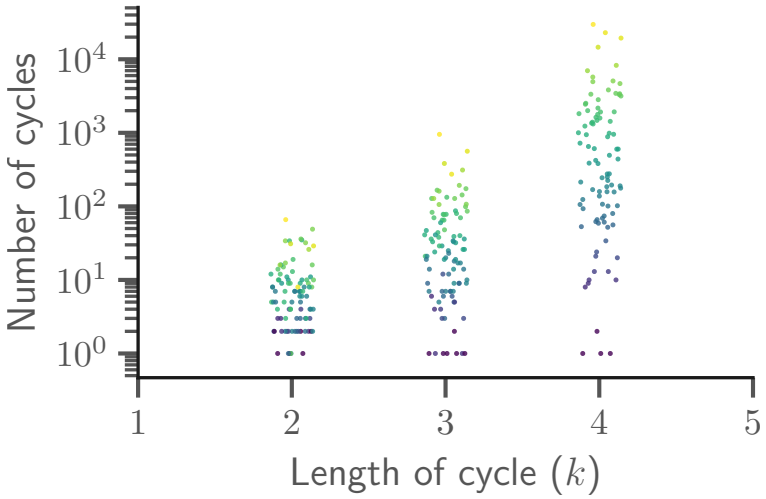

Figure 1: Number of cycles (log scale) by length for a sample of 100 random nodes. For each node in the sample, we have computed the number of cycles of length $k = 2, 3, 4$. Points representing the values for a single page are shifted on the x-axis by a random offset and colored with the same color. The color gradient depends on the value at $k = 4$.

6. Experimental Evaluation

This section is organized as follows: in Section (a) we describe the dataset that we have used for our experimental evaluation; Section (b) describes alternative approaches that we will use to compare to our proposed approach in addition to *Personalized PageRank*; Section (d) provides some example results and their qualitative description for each algorithm; in Section (e) we provide a detailed quantitative evaluation with three different evaluation measures based on different ground truth data. Finally, in Section (f) we compare the execution time of our proposed approach against the alternatives.

(a) Dataset Description

For our analysis, we used the *WikiLinkGraphs* dataset, consisting of the network of internal Wikipedia links for the 9 largest language editions [13]. The dataset has been developed by us and it is publicly available on Zenodo.[2] The graphs have been built by parsing each revision of each article to track links appearing in the main text, discarding links that were automatically inserted by templates. The dataset contains yearly snapshots of the network and spans 17 years, from the creation of Wikipedia in 2001 to March 1st, 2018. For the experiments in this paper we focused on the *WikiLinkGraphs* snapshot from English Wikipedia taken on March, 1st 2018. This graph has $N = 13{,}685{,}337$ nodes and $E = 163{,}380{,}007$ edges.[3]

Figure 1 presents the number of cycles by length for a sample of 100 nodes chosen randomly from our dataset. For each page we plot a triplet of points corresponding to the number of simple cycles of length $k = 2, 3$, and 4 respectively, that go through that node. We shift this triplet of points by a random offset along the horizontal axis for ease of reading.

[2] https://zenodo.org/record/2539424
DOI: 10.5281/zenodo.2539424.

[3] Wikipedia contains also special pages known as *redirects*, i.e. alternative articles titles. These pages appear in the graph as nodes with a single outgoing edge and typically no incoming edges. Our dataset consolidates alternative titles to the main one, but we still keep the redirect node; for this reason, the count of nodes in our graph differs from official count of the English Wikipedia.

Table 1 presents the top-10 pages by indegree and outdegree in the graph. We can see that indegree dominates outdegree by several orders of magnitude. This implies that ranking the top pages by degree (undirected) is *de facto* equivalent to ranking them by indegree. The difference between the top-1000 pages by indegree and the top-1000 pages by degree is of just 34 pages.

Table 2 presents the top-10 results by global *PageRank* score. Global *PageRank* and in-degree are highly correlated, regardless of the value of the damping parameter. When $\alpha = 0.85$ the two rankings have a Kendall correlation coefficient of $\tau = 0.60$ (over all pages with in-degree greater than zero, $n = 8{,}305{,}031$); if we limit the two rankings to the top 1,000 articles, they are still highly correlated with $\tau = 0.56$.

Table 1: Top-10 pages by indegree and outdegree over the most recent snapshot of the *WikiLinkGraphs* dataset (2018-03-01)

#	indegree		outdegree	
	article	degree	article	degree
1	United States	332,557	List of current U.S. state legislators	8,019
2	Animal	164,549	List of least concern birds	7,907
3	Association football	146,836	List of people from Illinois	7,827
4	India	126,107	List of birds of the world	6,849
5	World War II	124,806	List of stage names	6,677
6	Arthropod	122,742	List of cities, towns and villages in Kerman Province	5,839
7	Germany	121,705	List of film director and actor collaborations	5,804
8	Insect	118,628	Index of Telangana-related articles	5,747
9	Canada	115,779	Index of Andhra Pradesh-related articles	5,684
10	New York City	107,831	List of municipalities of Brazil	5,585

(b) Alternative Approaches

We describe briefly some alternative approaches that we will compare *CycleRank* with: beyond *Personalized PageRank*, we will consider the personalized versions of *CheiRank* and *2DRank*, which are all based on *Personalized PageRank*. From now on, for the sake of brevity, we will omit the specifier "personalized" when mentioning the algorithms, it will be clear from context if we refer to the regular, global algorithm or the personalized variant.

(i) CheiRank

CheiRank is a ranking algorithm first proposed by Chepelianskii [21], that consists in applying the *PageRank* algorithm on the transposed graph G^T, i. e. all link directions are inverted. This corresponds to transposing the adjacency matrix when computing *PageRank* and results in computing the conjugated Google matrix G^*.

CheiRank is analogous to *PageRank*, but it assigns a higher score to nodes with higher outdegree. In the Wikipedia dataset that we are using there are list articles that have several thousands outgoing links, as shown in Table 1. As expected, the articles with the highest global *CheiRank* score are list articles having high outdegree: out of the top 100 results by global *CheiRank* with $\alpha = 0.30$, 87 have the word `List`, `Lists`, or `Index` in the title. In the following,

Table 2: Top-10 pages by (global) *PageRank* over the most recent snapshot of the *WikiLinkGraphs* dataset (2018-03-01)

#	**PageRank,** $\alpha = 0.30$		**PageRank,** $\alpha = 0.85$	
	article	**score** ($\times 10^{-4}$)	**article**	**score** ($\times 10^{-4}$)
1	United States	4.64	United States	14.14
2	Animal	3.13	World War II	6.54
3	Arthropod	2.49	United Kingdom	6.18
4	Association football	2.45	Germany	5.57
5	Insect	2.42	The New York Times	5.27
6	Germany	2.16	Association football	5.25
7	List of sovereign states	2.11	List of sovereign states	5.23
8	India	2.03	Race and ethnicity in the United States Census	5.00
9	Moth	1.85	India	4.91
10	National Register of Historic Places	1.66	Canada	4.68

we are not showing results for *CheiRank* since it suffers from analogous limitations as *PageRank*, and it never resulted on-par with the most performing algorithms in our experiments.

(ii) 2DRank

2DRank combines *CheiRank* and *PageRank* [22]; it ranks all nodes in a graph, but it does not produce a score as *PageRank* or *CheiRank* do. Instead, given the rankings $\nu^{(\mathrm{PR})}$ and $\nu^{(\mathrm{ChR})}$ produced by *PageRank* and *CheiRank* respectively, *2DRank* takes the minimum position in which a given node appears in both ranking and builds a new ranking. This process can be visualized in the two-dimensional cartesian plane: xOy, we build a series of squares with one vertex in the origin, two sides formed by the cartesian axes and the other two drawn at integer values. Thus, the first square is identified by $(0;0)$, $(0;1)$, $(1;1)$, and $(1;0)$; the second by $(0;0)$, $(0;2)$, $(2;2)$, and $(2;0)$, and so on. By interpreting the position of an item in the *PageRank* (p) and *CheiRank* rankings (p^*) as the coordinates of a point $P(p, p^*)$, this point will fall on one of the edges of the squares drawn before. The position of a node in *2DRank* is given by assigning a progressive number to each item, starting from the points that lay on inner squares; if two points lay on the same square the algorithm chooses the one closest to either axis first.

(c) Implementation and Reproducibility

We implemented *CycleRank* in C++. For *Personalized PageRank* and *CheiRank* we used the `igraph` library,[4] *2DRank* was computed directly from *PageRank* and *CheiRank* results using a Python script. All code is available under an open-source license at: https://github.com/CycleRank/cyclerank.

(d) Qualitative Comparison

Tables 3 and 4 present a comparison between the top-10 results with the highest scores obtained by *CycleRank*, *PageRank* and *2DRank*, on the *WikiLinkGraphs* snapshot of the English Wikipedia of March 1st, 2018 with reference nodes *"Computer science"* and *"Freddie Mercury"*, respectively.

[4] https://igraph.org/c/

Table 3: Top-10 articles as ranked by *CycleRank*, *PageRank*, and *2DRank* with *"Computer science"* as reference node. The article *"Computer science"*, which would appear in the first position by definition, is omitted.

Computer science

#	**CycleRank,** $K = 3$		**CycleRank,** $K = 4$		**PageRank,** $\alpha = 0.30$		**PageRank,** $\alpha = 0.85$		**2DRank,** $\alpha = 0.30$	**2DRank,** $\alpha = 0.85$
	article	**score**	**article**	**score**	**article**	**score** ($\times 10^{-4}$)	**article**	**score** ($\times 10^{-4}$)	**article**	**article**
1	List of computer scientists	13.78	List of computer scientists	175.08	Computing	12.48	Mathematics	15.57	Association for Computing Machinery	Association for Computing Machinery
1	Algorithm	6.36	Algorithm	158.78	Computational science	12.07	Computer	14.42	Programming language	Programming language
3	Artificial intelligence	5.96	Artificial intelligence	154.30	Gottfried Wilhelm Leibniz	12.00	Computing	12.71	Theoretical computer science	Edsger W. Dijkstra
4	Theoretical computer science	5.36	Mathematics	134.13	Mathematics	10.92	Association for Computing Machinery	11.77	Artificial intelligence	Artificial intelligence
5	Mathematics	4.37	Programming language	108.42	Association for Computing Machinery	10.91	Gottfried Wilhelm Leibniz	11.71	Algorithm	Machine learning
6	Programming language	4.32	Theoretical computer science	100.13	Algorithm	10.58	Computational science	11.30	Programming language theory	Algorithm
7	List of pioneers in computer science	4.08	List of pioneers in computer science	88.08	Artificial intelligence	9.97	Algorithm	11.21	Edsger W. Dijkstra	Programming language theory
8	List of important publications in computer science	4.02	Alan Turing	79.11	IBM	9.93	United States	10.23	List of computer scientists	Compiler
9	Edsger W. Dijkstra	3.82	Logic	75.90	Computational complexity theory	9.91	World War II	10.04	Data science	Theoretical computer science
10	Alan Turing	3.67	Outline of software engineering	73.35	Logic	9.84	IBM	9.60	Machine learning	List of pioneers in computer science

Table 4: Top-10 articles as ranked by *CycleRank*, Personalized PageRank, and 2Drank with *"Freddie Mercury"* as reference node. The article *"Freddie Mercury"*, which would appear in the first position by definition, is omitted.

Freddie Mercury

#	**CycleRank,** $K = 3$		**CycleRank,** $K = 4$		**PageRank,** $\alpha = 0.30$		**PageRank,** $\alpha = 0.85$		**2DRank,** $\alpha = 0.30$	**2DRank,** $\alpha = 0.85$
	article	**score**	**article**	**score**	**article**	**score** ($\times 10^{-4}$)	**article**	**score** ($\times 10^{-4}$)	**article**	**article**
1	Queen (band)	13.78	Queen (band)	484.34	Queen (band)	10.79	Queen (band)	13.40	Queen (band)	Queen (band)
2	Brian May	6.41	Brian May	147.90	The Freddie Mercury Tribute Concert	8.92	London	12.07	Roger Taylor (Queen drummer)	Brian May
3	Roger Taylor (Queen drummer)	4.72	Elton John	97.48	HIV/AIDS	8.78	United States	11.51	Brian May	Roger Taylor (Queen drummer)
4	John Deacon	3.52	Roger Taylor (Queen drummer)	92.80	Queen II	8.65	Rock music	8.99	Made in Heaven	Queen II
5	Made in Heaven	3.37	Michael Jackson	73.33	Mercury Phoenix Trust	8.52	HIV/AIDS	8.99	Queen II	Made in Heaven
6	We Will Rock You (musical)	3.17	John Deacon	64.73	Middlesex	8.48	The Freddie Mercury Tribute Concert	8.73	John Deacon	John Deacon
7	Elton John	3.12	Live Aid	61.51	Parsi	8.45	Roger Taylor (Queen drummer)	8.51	We Will Rock You (musical)	Bohemian Rhapsody
8	The Freddie Mercury Tribute Concert	3.07	Bohemian Rhapsody	60.24	Zoroastrianism	8.39	United Kingdom	8.30	Bohemian Rhapsody	We Will Rock You (musical)
9	Bohemian Rhapsody	3.02	Brit Awards	53.69	Good Old-Fashioned Lover Boy	8.35	BBC	8.24	Greatest Hits (Queen album)	Greatest Hits (Queen album)
10	Greatest Hits (Queen album)	2.97	The Freddie Mercury Tribute Concert	48.50	Millennium stamp	8.33	Queen II	8.07	Panchgani	Charles Messina

These results highlight the limitations of *Personalized PageRank* described in Section 3: in the top positions in the rankings produced with $\alpha = 0.85$, we see articles such as *"United States"* and *"World War II"*; these articles act as attractors for the unconstrained random walks of *PageRank* since they have a very high in-degree and have among the highest values of the *PageRank* score in the overall network, as shown in Tables 1 and 2. Indeed, they are respectively in 1st (*"United States"*) and 2nd (*"World War II"*) position in the overall *PageRank* ranking for the network. This problem is only partially mitigated by lowering the damping factor to $\alpha = 0.30$.

However, there are much fewer paths that connect these articles back to the reference nodes. As a result, these articles appear in much lower positions in the ranking produced by the *CycleRank* algorithm: for example, for *"Computer science"* they appear respectively in 400th (*"United States"*), and 172th (*"World War II"*) position. In this way, *CycleRank* leaves space to articles whose content is more strongly associated with the reference topic to appear at higher positions in the ranking.

Similarly, the *Personalized PageRank* results for *"Freddie Mercury"* suffer from the same problem: *"United States"*, *"London"* (17th position in the global ranking), *"United Kingdom"* (3rd position in the global ranking) and *"BBC"* (53rd position in the global ranking) all appear in the top-10 results for the *Personalized PageRank* with $\alpha = 0.85$. This bias is only partially resolved by lowering the damping factor.

2DRank seems to mostly solve this issue, but still includes spurious results such as *"Charles Messina"* and *"Panchgani"*, that are only partially related to *"Freddie Mercury"*.

A more extended qualitative comparison can be found in [26].

(e) Quantitative Comparison

To compare our proposed approach against existing algorithms, we need a way to evaluate how good a ranking is with respect to some ground truth. In general, we cannot directly compare the ranking functions rf_r, because they may vary wildly in absolute values; furthermore, some algorithms we will compare to do not define a ranking function but just produce the final ranking.

We provide three different comparison strategies that we encapsulate in three different measures. Each measure is based on a suitable dataset that we use as a ground truth against which we evaluate the performance of each algorithm. At high level, we want to evaluate the following three facets of each ranking algorithm:

(i) to what extent it is able to maintain the relative ranking of most-clicked links from a given article (*ClickStream* evaluation);
(ii) to what extent it is able to rank in the top positions articles highlighted by editors in the "See Also" section (*See-Also* evaluation).
(iii) to what extent it tends to give prominence to global "superstars", i. e. nodes which are very popular in the overall network as measured by their high indegree (*"Indegree"* evaluation).

Our evaluation measures are based on the ones used in the information retrieval literature. In particular, we follow in the footstep of Schwarzer and collaborators [19] who have also used *ClickStream* and *See-Also* for evaluating performance across a large set of topics.

In the following subsections, we describe in detail each measure, the dataset used as ground truth and the results of the experiments we performed. We also present examples to illustrate qualitatively the results of each experiment.

(i) ClickStream Evaluation

The idea of this measure is to test the ability of each algorithm to maintain the relative relevance of a set of topics with respect to the *ClickStream* dataset [27], which we use as a ground truth. In other words, interpreting clicks on links by Wikipedia readers as a measure of the relative importance of each link in an article, we aim to measure whether the algorithms are able to maintain this relative ranking.

We have chosen the February 2018 release [5] of the dataset because it is the closest in time to our WikiLinkGraphs snapshot. This dataset contains counts of `(source, target)` article pairs extracted from the request logs made to Wikipedia's servers over one month. This data reflects the number of times a Wikipedia visitor has reached the `target` article from the `source` article. The fact that a given `(source, target)` pair appears in the clickstream implies the existence of a link in the `source` page pointing to the `target` page; these links may appear as *wikilinks* in the article source or come from templates. Note that `(source, target)` pairs with a count of 10 or fewer observations are not present in the dataset. In this way, this data provides an aggregated view on how Wikipedia articles are reached by users and what links they click on, producing a weighted network of articles, where each edge weight corresponds to how often people navigate from one page to another.

The *ClickStream* dataset also contains special sources to represent, for example, pages in other Wikimedia projects or external search engines;[6] we filter those out. The dataset in total comprises over 25M pairs, of which over 15.4M are links between pages.

From the *ClickStream* data we can derive an ordered list of articles, which we can consider as a ranking: our evaluation strategy consists in computing Kendall's rank correlation coefficient between the *ClickStream* ranking and the ranking of the same pages produced by the algorithms under consideration. We formalize this evaluation strategy as follows: let $\mathcal{C}$ be the *ClickStream* dataset, i.e. a set of triplets: (v_s, v_t, c) where $v_s, v_t \in V$ are respectively the source and target articles and $c \in \mathbb{N}, c \geq 10$ is the count for the pair (v_s, v_t); we define $W_r \subseteq V$ as the set of nodes that appear in the *ClickStream* dataset with source r, i. e., for some count c:

$$W_r = \{w \in V | (r, w, c) \in \mathcal{C}\}.$$

We then use the counts in the *ClickStream* dataset to define a $rf_r^{\mathcal{C}}$ over the set W_r. Given a target $w \in W_r$ if the count for the pair (r, w) is c, i. e. if $(r, w, c) \in \mathcal{C}$ then:

$$rf_r^{\mathcal{C}}(w) = c$$

The ranking function defined above produces a ranking $\nu_r^{\mathcal{C}} = [w_1, w_2, \ldots, w_q]$ of the nodes in W_r. Ties are broken at random. The ranking will be the ground truth for evaluating the performance of each algorithm for node r.

Let ν_r be a ranking of the nodes in V produced by one of the algorithms under consideration when r is the reference node. We restrict this ranking to only the pages that appear in the *ClickStream* data $\nu_r|_{W_r}$ and then we build a list of q pairs from the rankings: $[(v_1, w_1), (v_2, w_2), \ldots, (v_q, w_q)]$.

Given two pairs (v_i, w_i) and (v_j, w_j) where $i < j$, these pairs are said to be concordant if the ranks for both elements agree: i. e., if both $v_i > v_j$ and $w_i > w_j$, or analogously if $v_i < v_j$. If $v_i = v_j$ or $w_i = w_j$ two pairs are neither concordant nor discordant. Otherwise they are discordant.

The quality of the ranking ν_r, is then defined as Kendall's rank correlation coefficient:

$$\tau(\nu_r^{\mathcal{C}}, \nu_r) = \frac{\pi_+ - \pi_-}{\binom{q}{2}}$$

where π_+ and π_- are the number of concordant and discordant pairs, respectively. We say that a ranking ν_r^1 is better than a ranking ν_r^2 if its rank correlation with the *ClickStream* ranking is higher: $\tau(\nu_r^1) > \tau(\nu_r^2)$.

Table 5 presents an example of how this evaluation metric works for the article *"Computer science"*: the table shows the *ClickStream* data and the induced ranking $\nu_r^{\mathcal{C}}$, as well as the rankings produced by the *CycleRank* ($\nu_r^{(\mathrm{CR})}$), *PageRank* ($\nu_r^{(\mathrm{PR})}$), and *2DRank* ($\nu_r^{(\mathrm{2D})}$) algorithms over the same articles. Regardless of the absolute position of these articles, we measure how these rankings

[5] https://dumps.wikimedia.org/other/clickstream/2018-02/

[6] More precisely, the dataset contains the counts of `(referer, resource)` pairs extracted from Wikipedia's webserver logs. A `referer` is an HTTP header field that identifies the webpage that linked to the resource being requested, a `resource` is the target of the request.

Table 5: *ClickStream* data for the article *"Computer science"* (c is the click count, $\nu_r^{\mathcal{C}}$ is the ranking induced by the count) and rankings produced by *CycleRank* with $K = 3$ (CR) and *PageRank* with $\alpha = 0.30$ (PR) after filtering. The Kendall correlation coefficients between *ClickStream* and the rankings produced by the algorithms presented in the table are computed only over the 10 items displayed.

Computer science					
article (W_r)	c	$\nu_r^{\mathcal{C}}$	$\nu_r^{(\mathrm{CR})}$	$\nu_r^{(\mathrm{PR})}$	$\nu_r^{(\mathrm{2D})}$
Computation	1371	1	56	65	77
Algorithm	876	2	2	6	5
Programming language theory	794	3	17	63	6
Computer graphics (computer science)	648	4	43	134	31
Computational complexity theory	647	5	33	9	108
Human–computer interaction	550	6	47	68	50
Computer scientist	480	7	59	20	62
Outline of computer science	452	8	204	298	173
Computer programming	451	9	62	18	160
Programming language	414	10	6	12	2
$\tau(\nu_r^{\mathcal{C}}, \nu_r)$			0.3333	−0.0222	0.2444

agree with the one given by the *ClickStream* data; a negative value means that the ranking is discordant with the *ClickStream* ranking.

Figures 2 and 3 present the results of the *ClickStream* evaluation over a sample of 1,000 random articles. We have built this sample by selecting random Wikipedia articles that have at least 5 entries in the *ClickStream* data, i. e. they have at least 5 links to other Wikipedia articles. In the figures, the (x, y) coordinates of each point are the values of Kendall's rank correlation coefficient with *ClickStream* data for *PageRank* and *CycleRank* (Figure 2); and for *2DRank* and *CycleRank* (Figure 3). Thus, if points have their y coordinate greater than the x coordinate, i.e. they are above the dashed axis $Y = x$ in the figure, it means that *CycleRank* is outperforming the other approach for that article. This is the case both for the comparison with *PageRank* in Figure 2, where 68.8% of articles are above the axis, and with a smaller margin for the comparison with *2DRank* in Figure 3, where 50.2% of articles are above the axis.

Table 6 presents the results of the *ClickStream* evaluation for different values of the algorithms' parameters. Each cell reports two percentage values, representing respectively the fraction of articles for which the chosen baseline produced a ranking with higher/lower correlation with *ClickStream* than *CycleRank*. This corresponds to the proportion of points below and above the $Y = x$ axis in Figures 3 and 3. Asterisks are used to indicate the statistical significance of the difference according to a paired t-test. The best results are obtained by *CycleRank* with $K = 3$; in the comparison with *2DRank* the difference is not statistically significant.

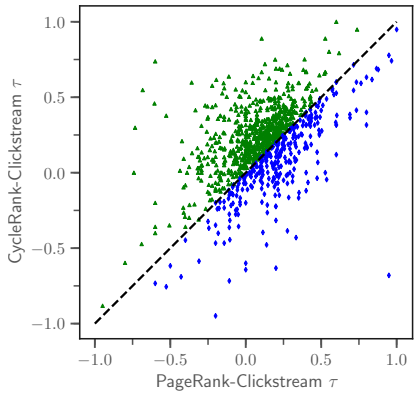

Figure 2: Comparison of the Kendall τ correlation coefficients of the *ClickStream* ranking with the rankings produced by *PageRank* (x coordinate) and *CycleRank* (y coordinate) over a sample of 1000 random articles. If for a given article $y > x$ then the correlation between the *CycleRank* and *ClickStream* rankings is higher than the correlation between the *PageRank* and *ClickStream* rankings (green triangles), if $y \leq x$ is vice-versa (blue diamonds).

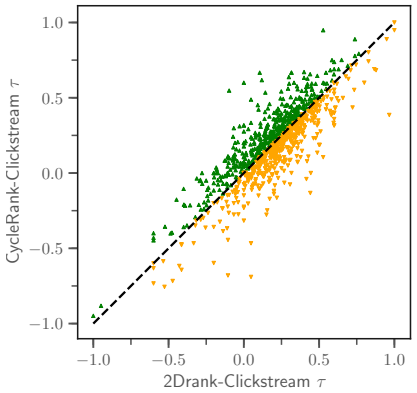

Figure 3: Comparison of the Kendall τ correlation coefficients of the *ClickStream* ranking with the rankings produced by *2DRank* (x coordinate) and *CycleRank* (y coordinate) over a sample of 1000 random articles. If for a given article $y > x$ then the correlation between the *CycleRank* and *ClickStream* rankings is higher than the correlation between the *2DRank* and *ClickStream* rankings (green triangles), if $y \leq x$ is vice-versa (blue diamonds).

Table 6: Results of the *ClickStream* evaluation over a sample of 1,000 random articles. The two percentages reported in each cell represent the proportions of articles for which the correlation is higher for the algorithm indicated by the row and by the column, respectively. The amount missing to sum to 100% corresponds to cases of equal correlation. Asterisks indicate the significance level according to the p-values obtained from paired t-test: $^{*}: p \leq 0.05$; $^{**}: p \leq 0.001$; $^{***}: p \leq 0.0001$.

algorithm		CycleRank	
	parameters	$K = 3$	$K = 4$
PageRank	$\alpha = 0.30$	$30.2/68.8^{***}$	$37.6/59.0^{***}$
	$\alpha = 0.85$	$33.4/65.5^{***}$	$36.2/59.7^{***}$
2Drank	$\alpha = 0.30$	$48.2/50.2$	$58.4/37.4^{***}$
	$\alpha = 0.85$	$48.2/50.4$	$56.9/37.7^{***}$

(ii) *See-Also* Evaluation

We measure the ability of an algorithm to identify relevant articles by using links in the *See-Also* section of a Wikipedia article as a ground truth. Following Wikipedia policies [28], the section *See-Also* contains a list of internal links to related Wikipedia articles. These lists may be ordered logically, chronologically or alphabetically, and there is no guarantee that the same criterion is used across multiple pages. For this reason, we treat these lists as non-ordered, that is we do not

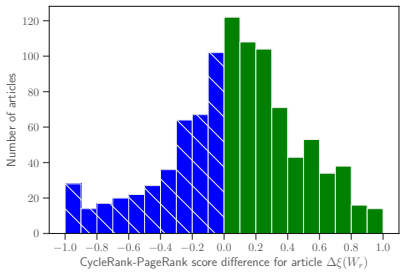

Figure 4: Distribution of $\Delta\xi(W_r)$ between *CycleRank* and *PageRank*. When values are positive (solid green bars) *CycleRank* is able to find *See-Also* articles in a higher position than Personalized Pagerank for a given article; when values are negative (blue bars with with white hatch) is vice-versa.

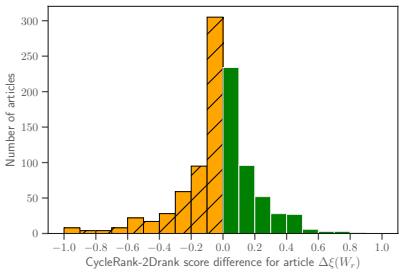

Figure 5: Distribution of $\Delta\xi(W_r)$ between *CycleRank* and *2DRank*. When values are positive (solid green bars), *CycleRank* is able to find *See-Also* articles in a higher position than *2DRank* for a given article; when values are negative (orange bars with with black hatch) is vice-versa.

treat the pages listed in these sections as being ranked by relevance, but just as a set of related pages.

More formally, let $W_r \subseteq V$ be a set of ground-truth nodes that are relevant with respect to a reference node r, and let ν_r be a ranking of the nodes in V. The quality of the ranking ν_r is defined as

$$\xi(\nu_r, W_r) = \sum_{w \in W} \xi(\nu_r, w)$$

where

$$\xi(\nu_r, w) = \frac{1}{i} \quad \text{if } w = v_i \in \nu_r \wedge w \in W$$

We say that a ranking ν_r^1 is better than a ranking ν_r^2 if $\xi(\nu_r^1) > \xi(\nu_r^2)$.

Table 7 presents, as an example, the results of the *See-Also* evaluation for the article *"Computer Science"*: the first column lists the name of the articles appearing in the *See-Also* section of the article *"Computer science"* W_r; the second, third and fifth columns show the position of each article in the ranking produced by *CycleRank*, *PageRank*, and *2DRank* respectively; the fourth and sixth columns show the difference in evaluation score for each article between *CycleRank* and *Personalized PageRank* $\Delta\xi(\text{CR} - \text{PR})$, and *CycleRank* and *2DRank* $\Delta\xi(\text{CR} - \text{2D})$. When *CycleRank* ranked a page in a higher position than the other approach this difference is positive, otherwise it is negative.

Figures 4 and 5 show the distribution of the differences in evaluation score over the same sample of 1,000 articles used above. Figure 4 compares *CycleRank* and *Personalized PageRank*, i. e. the plot is the distribution $\Delta\xi(\text{CR} - \text{PR})$, while Figure 5 presents analogous results for *CycleRank* and *2DRank*, $\Delta\xi(\text{CR} - \text{PR})$.

Table 8 presents the results of the *See-Also* evaluation over a sample of 1,000 random articles. We have built this sample with the following characteristics: we selected Wikipedia articles that have at least 3 links to other existing Wikipedia articles[7], in this way we ensure that the pages used in the sample have enough links.

As the table shows, the best performance is achieved by *CycleRank* with $K = 3$, with results that significantly outperform all the other algorithms. The second-best algorithm is *2DRank* with $\alpha = 0.30$. These results confirm that in the context under analysis it is important not only to account for both incoming and outgoing links, as *CycleRank* and *2DRank* do, but also to limit longer paths: indeed the best results are found for lower values of K and of α, respectively.

[7] Even if it is discouraged by Wikipedia policies, in principle a Wikipedia editor could insert in the *See-Also* section a link to a non-existing article.

Table 7: The first 10 articles appearing in the *See-Also* section of the *"Computer science"* article. We use them to compare *CycleRank* with $K = 3$ (CR), *PageRank* with $\alpha = 0.30$ (PR), and *2DRank* with $\alpha = 0.30$ (2D). For each article, the table reports the position in which it appears in the ranking produced by each algorithm, and the corresponding difference in scores $\Delta\xi$. The $\sum \Delta\xi(W_r)$ is calculated only over the 10 items displayed.

Computer science					
article (W_r)	$\nu_r^{(CR)}$	$\nu_r^{(PR)}$	$\Delta\xi$ CR-PR $(\times 10^{-4})$	$\nu_r^{(2D)}$	$\Delta\xi$ CR-2D $(\times 10^{-4})$
Academic genealogy of computer scientists	13	220	723.8	26	384.6
Association for Computing Machinery	16	6	−1041.7	2	−4375.0
Computer Science Teachers Association	207	231	5.0	402	23.4
Engineering informatics	447	228	−21.5	399	−2.7
Informatics	70	106	48.5	87	27.9
List of academic computer science departments	74	232	92.0	7862	133.9
List of computer scientists	2	110	4909.1	9	3888.9
List of important publications in computer science	9	167	1051.2	18	555.6
List of pioneers in computer science	8	16	625.0	538	1231.4
List of unsolved problems in computer science	92	148	41.1	41	−135.2
Outline of software engineering	12	217	787.3	25	433.3
Technology transfer in computer science	206	223	3.7	380	22.2
Turing Award	14	49	510.2	17	12.6
$\sum \Delta\xi(W_r)$			7733.8		2314.4

Table 8: Results of the *See-Also* evaluation over a sample of 1,000 random articles for *CycleRank*, *PageRank* and *2DRank* for different values of their parameters. Each cell reports the mean of the difference between the ξ obtained over the sample of 1,000 articles. Positive (negative) values indicate higher (lower) performance for *CycleRank*. Asterisks indicate the significance level according to the p-values obtained from paired t-test: $^{*}: p \leq 0.05$; $^{**}: p \leq 0.001$; $^{***}: p \leq 0.0001$.

algorithm		CycleRank	
	parameters	$K = 3$	$K = 4$
PageRank	$\alpha = 0.30$	0.062^{***}	0.021
	$\alpha = 0.85$	0.111^{***}	0.069^{***}
2Drank	$\alpha = 0.30$	0.017^{***}	-0.065^{***}
	$\alpha = 0.85$	0.063^{***}	-0.045^{***}

(iii) Indegree Evaluation

We measure the extent to which an algorithm tends to give prominence to global "superstars", i. e. nodes which are very popular in the overall network as measured by their high indegree. In this section we use the same measure ξ that we have used in the previous *See-Also* evaluation, with two modifications. First, we use the top-100 articles by indegree as test set W_r and in this case lower is better, as it implies a lower presence of global hubs at the top of the ranking produced: we

consider a ranking ν_r^1 better than a ranking ν_r^2 if $\xi(\nu_r^1) < \xi(\nu_r^2)$. Second, as *PageRank* and *2DRank* produce longer rankings, for a fair comparison we cut-off all rankings at 1,000 results.

Table 9: Positions in which the top-100 articles by indegree appear in the rankings produced by *CycleRank* (top), *PageRank* (middle), and *2DRank* (bottom) with *"Freddie Mercury"* as reference node and their score $\xi(\nu_r, w)$. The ranking for *PageRank* and *2DRank* are limited to the top-1000 positions. The $\sum \xi(W_r)$ is calculated only over the 10 items displayed.

Freddie Mercury

ν^i	**CycleRank,** $K = 3$		
	article	$\nu_r^{(CR)}$	$\xi(\nu_r^{(CR)}, w)$ $(\times 10^{-4})$
13	London	88	113.64
63	BBC	364	27.47
87	Rock music	55	181.82
$\sum \xi(\nu_r^{(CR)}, W_r)$			322.93
	PageRank, $\alpha = 0.30$		
	article	$\nu_r^{(PR)}$	$\xi(\nu_r^{(PR)}, w)$ $(\times 10^{-4})$
1	United States	363	27.55
4	India	383	26.11
9	Canada	449	22.27
10	New York City	367	27.25
13	London	12	833.33
17	Italy	761	13.14
18	Iran	613	16.31
19	Japan	565	17.70
20	The New York Times	364	27.47
21	California	495	20.20
29	Spain	879	11.38
$\sum \xi(\nu_r^{(PR)}, W_r)$			1042.71
	2DRank, $\alpha = 0.30$		
	article	$\nu_r^{(2D)}$	$\xi(\nu_r^{(CR)}, w)$ $(\times 10^{-4})$
13	London	603	16.58
54	The Guardian	569	17.57
87	Rock music	426	23.47
$\sum \xi(\nu_r^{(2D)}, W_r)$			57.63

Table 9 presents the position in which the top-100 articles by indegree appear in the top-1000 positions of the rankings produced *Personalized PageRank* (top), *2DRank* (middle), and *CycleRank* (bottom) with *"Freddie Mercury"* as reference node.

Figure 6 shows the results of the indegree evaluation for *PageRank*, *2DRank*, and *CycleRank* on a sample of 1,000 random articles, taking the top-N results for each article, with values on N ranging until 1,000. We see that *CycleRank* is able to obtain a lower score meaning that it includes fewer pages with high indegree in high position in the rankings it produces.

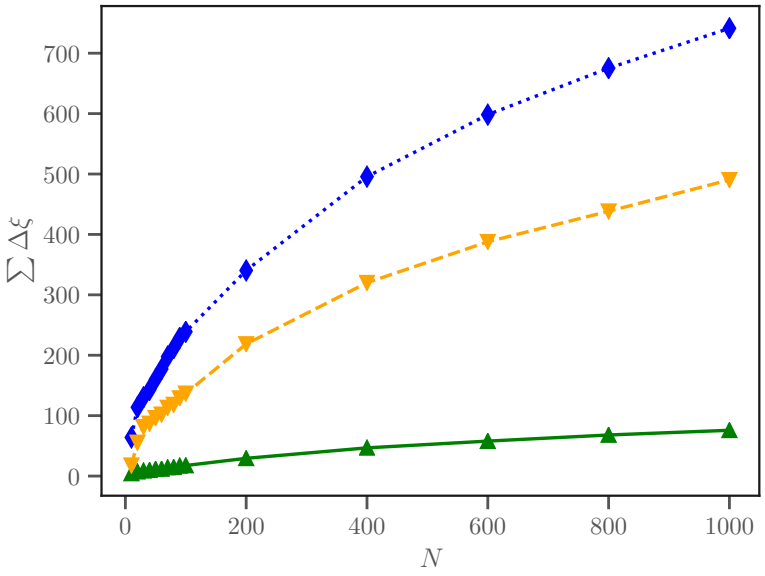

Figure 6: Indegree evaluation scores for *PageRank* (dotted blue line with diamond markers), *2DRank* (dashed orange line with triangle-down markers), and *CycleRank* (straight green line with triangle-up markers) taking the top-N articles by indegree. A lower score means the ranking produced by the algorithm is more robust to global hubs.

Table 10: Execution time comparison of *CycleRank*, *PageRank*, *CheiRank*, and *2DRank* for different values of the parameters: maximum cycle length K for *CycleRank*, and damping parameter α for *PageRank*, *CheiRank*, and *2DRank*. All times are expressed in seconds.

algorithm	parameter	time (s)
CycleRank	$K = 3$	4 ± 1
	$K = 4$	10 ± 16
PageRank	$\alpha = 0.30$	260 ± 13
	$\alpha = 0.85$	928 ± 80
CheiRank	$\alpha = 0.30$	258 ± 20
	$\alpha = 0.85$	374 ± 50
2DRank	$\alpha = 0.30$	> 518
	$\alpha = 0.85$	> 1302

(f) Performance Analysis

Finally, Table 10 evaluates the performance of *CycleRank* with respect to the alternative approaches. Times are computed by averaging over the sample of 1,000 articles used in the *See-Also* evaluation. Experiments were performed on a HPC cluster, on nodes equipped with Intel Xeon E5-2650V3 (10 core) processors and 256 GB of RAM. Each job computed *CycleRank*, *Personalized PageRank*, or *2DRank* for one given seed node r using only one core and one processor at a time.

Results presented in Table 10 do not take into account the time needed to read the input graph, which required 60 seconds on average (a value much larger than the time need by *CycleRank* to execute). Execution times for *2DRank* are not obtained directly, but by summing the execution times of *Personalized PageRank* and *CheiRank*.

Computing *CycleRank* is two orders of magnitude faster than computing *Personalized PageRank* and *CheiRank*. Since our proposed approach is based on enumerating all cycles going through the reference node, in the worst case–a complete graph–the computational complexity increases exponentially with cycle length, making its computation challenging for higher values of K in very dense graphs. However, we have shown that *CycleRank* can produce good results for small values of K and has a significant time advantage with respect to *Personalized PageRank* and *2DRank*.

7. Conclusions

This paper introduces *CycleRank*, a novel algorithm based on cyclic paths that can be used to assign relevance scores to nodes in a directed graph. Given a reference node, the algorithm finds all simple cycles that go through this node and assigns a score to each node that belongs to these cycles. The algorithm is characterized by one parameter, which is the maximum cycle length to be considered.

We have performed an extensive comparison between *CycleRank*, *PageRank*, and *2DRank*, based on three quantitative measures. The first experiment, based on the *ClickStream* dataset, has shown that the rankings produced by *CycleRank* align better with readers' behavior, with *2DRank* obtaining comparable results. The second experiment, based on *See-Also* links that appear in Wikipedia articles, has shown that *CycleRank* is able to rank related articles in a higher position. The third experiment, based on pages with high in-degree, has shown that *CycleRank* is more robust to the influence of network hubs. Furthermore, we have shown that our algorithm is faster than the alternatives, offering order-of-magnitude speed-ups with respect to library implementation of *Personalized PageRank*.

In other words, *CycleRank* is a viable alternative to *Personalized PageRank*, especially in the case of graphs where the role of inlinks and outlinks is comparable; our experiments on the Wikipedia link graph have shown that *CycleRank* achieves better performance both in terms of accuracy and efficiency.

The best results in all the experiments were obtained by *CycleRank* with $K = 3$, i.e. limiting the maximum length of cycles to 3. This resonates with what is seen for *PageRank* and *2DRank* in this context, where smaller values of the damping factor α achieve better results, giving lower scores to nodes that are further from the reference node. This could be associated with the high density of the Wikipedia link graph: as soon as a random path gets further away from the reference node, the influence of hubs and of denser areas of the network gets higher. Although *CycleRank* is better than *PageRank* and *2DRank* at limiting this issue, with higher values of K it is still affected.

The algorithm was developed for the context of the Wikipedia link network, but we believe it has potential to be employed in a variety other contexts. Contexts like knowledge bases or the Web of Data [29] can be seen as analogous to the one considered, with links between entities representing semantic relations to be considered in both directions. In social media, explicit links such as friendship or followership, and implicit links based of interactions (e.g. retweets or mentions) are often used to model influence between users, and to recommend new contacts [30]. While current models mostly consider influence as uni-directional, *CycleRank* would allow for the identification of relevance with respect to a specific user accounting not only for their interests and preferences, but also for the attention they receive from others.

We believe that the choice of an appropriate scoring function merits further analysis in future research. While we have empirically validated the results on our *WikiLinkGraphs* dataset for other types of scoring functions, in this work for reasons of space we have presented only a simple exponentially-decaying scoring function, for which we obtained the best results. Many variations and extensions could be explored, we expect that linear or quadratic functions would be more apt for sparser networks, while more skewed functions could be used for denser networks; in the latter case it could be worth to explore the usage of exponential functions with a higher base than the Euler's number, e.g. the average degree of the network, as a proxy for the number of cycles.

We have assumed the starting point for the algorithm to be a single reference node. However, as in the case of *Personalized PageRank*, it would be possible to take a group of articles as seed. Then, one could count all paths from any node in the seed to any other node in the seed. Another possible variant would be to specify two different nodes (or groups of nodes) as source and target and to consider all paths from the source to the target within K steps. In this way, the measure would not represent the relevance of other nodes with respect to a reference node but to the (directed) relationship between two nodes or groups of nodes. This measure would help to answer questions such as: "Which are the most relevant concepts connecting Artificial Intelligence and Human rights, and which are the most relevant concepts on the other way round"?

We believe that *CycleRank* provides a foundation that could be further explored to provide a family of algorithms adapted for different graphs and use cases. The suitability of different solutions could also be studied with respect to the structural properties of the network under analysis, such as its link density or clustering coefficient.

8. Acknowledgements

The authors would like to thank the participants to the WikiWorkshop 2019 for their precious feedback on a preliminary version of the *CycleRank* algorithm.

Data Accessibility. The datasets and the code used in this article are available. The following datasets have been used: (a) Wikimedia XML dumps are available at: https://dumps.wikimedia.org/; (b) The *WikiLinkGraphs* dataset is available on Zenodo at: https://zenodo.org/record/2539424; (c) The ClickStream dataset is available at: https://dumps.wikimedia.org/other/clickstream/. The code is available on GitHub at: https://github.com/CycleRank/cyclerank.

Authors' Contributions. CC conceived the algorithm, designed and performed the experiments and wrote the manuscript. DL conceived the study, helped design the experiments and helped draft the manuscript. AM supervised the work and helped draft the manuscript. All authors gave final approval for publication and agree to be held accountable for the work performed therein.

Funding. CC and DL have been supported by the European Union's Horizon 2020 research and innovation programme under the EU ENGINEROOM project, with Grant Agreement nº 780643.

References

1. Alexa Internet, Inc.. 2019 The top 500 sites on the web. https://www.alexa.com/topsites. [Online; accessed 13-March-2019].
2. Borra E, Weltevrede E, Ciuccarelli P, Kaltenbrunner A, Laniado D, Magni G, Mauri M, Rogers R, Venturini T. 2015 Societal Controversies in Wikipedia Articles. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, Seoul, Republic of Korea, April 18-23, 2015* pp. 193–196.
3. Mesgari M, Okoli C, Mehdi M, Nielsen FÅ, Lanamäki A. 2015 "The sum of all human knowledge": A systematic review of scholarly research on the content of W ikipedia. *Journal of the Association for Information Science and Technology* **66**, 219–245.
4. Navigli R, Ponzetto SP. 2012 BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* **193**, 217–250.
5. Yeh E, Ramage D, Manning CD, Agirre E, Soroa A. 2009 WikiWalk: random walks on Wikipedia for semantic relatedness. In *Proceedings of the 2009 workshop on graph-based methods for natural language processing* pp. 41–49. Association for Computational Linguistics.
6. Aragon P, Laniado D, Kaltenbrunner A, Volkovich Y. 2012 Biographical social networks on Wikipedia: a cross-cultural study of links that made history. In *Proceedings of the eighth annual international symposium on Wikis and open collaboration* p. 19. ACM.
7. Eom YH, Aragón P, Laniado D, Kaltenbrunner A, Vigna S, Shepelyansky DL. 2015 Interactions of cultures and top people of Wikipedia from ranking of 24 language editions. *PloS one* **10**, e0114825.

8. Capocci A, Servedio VD, Colaiori F, Buriol LS, Donato D, Leonardi S, Caldarelli G. 2006 Preferential attachment in the growth of social networks: The internet encyclopedia Wikipedia. *Physical review E* **74**, 036116.
9. West R, Leskovec J. 2012 Human wayfinding in information networks. In *Proceedings of the 21st international conference on World Wide Web* pp. 619–628. ACM.
10. Lin Y, Yu B, Hall A, Hecht B. 2017 Problematizing and addressing the article-as-concept assumption in Wikipedia. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* pp. 2052–2067. ACM.
11. Markusson N, Venturini T, Laniado D, Kaltenbrunner A. 2016 Contrasting medium and genre on Wikipedia to open up the dominating definition and classification of geoengineering. *Big Data & Society* **3**, 2053951716666102.
12. Pentzold C, Weltevrede E, Mauri M, Laniado D, Kaltenbrunner A, Borra E. 2017 Digging Wikipedia: The Online Encyclopedia as a Digital Cultural Heritage Gateway and Site. *Journal on Computing and Cultural Heritage (JOCCH)* **10**, 5.
13. Consonni C, Laniado D, Montresor A. 2019 WikiLinkGraphs: A complete, longitudinal and multi-language dataset of the Wikipedia link networks. In *Proceedings of the International AAAI Conference on Web and Social Media* vol. 13 pp. 598–607.
14. Page L, Brin S, Motwani R, Winograd T. 1999 The PageRank citation ranking: Bringing order to the web.. Technical report Stanford InfoLab.
15. Gleich DF, Constantine PG, Flaxman AD, Gunawardana A. 2010 Tracking the random surfer: empirically measured teleportation parameters in PageRank. In *Proceedings of the 19th international conference on World wide web* pp. 381–390. ACM.
16. Boldi P, Santini M, Vigna S. 2005 PageRank as a function of the damping factor. In *Proceedings of the 14th international conference on World Wide Web* pp. 557–566. ACM.
17. Labhishetty S, Siddiqa A, Nagipogu R, Chakraborti S. 2017 WikiSeeAlso: Suggesting Tangentially Related Concepts (See also links) for Wikipedia Articles. In *International Conference on Mining Intelligence and Knowledge Exploration* pp. 274–286. Springer.
18. Ollivier Y, Senellart P. 2007 Finding related pages using Green measures: An illustration with Wikipedia. In *AAAI* vol. 7 pp. 1427–1433.
19. Schwarzer M, Schubotz M, Meuschke N, Breitinger C, Markl V, Gipp B. 2016 Evaluating link-based recommendations for Wikipedia. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries* pp. 191–200. ACM.
20. Kamps J, Koolen M. 2009 Is wikipedia link structure different?. In *Proceedings of the second ACM international conference on Web search and data mining* pp. 232–241. ACM.
21. Chepelianskii AD. 2010 Towards physical laws for software architecture. *arXiv:1003.5455*.
22. Zhirov A, Zhirov O, Shepelyansky D. 2010 Two-dimensional ranking of Wikipedia articles. *The European Physical Journal B* **77**, 523–531.
23. Torres L, Suárez-Serrato P, Eliassi-Rad T. 2019 Non-backtracking cycles: length spectrum theory and graph mining applications. *Applied Network Science* **4**, 41.
24. Brouwer AE, Haemers WH. 2011 *Spectra of graphs*. Springer Science & Business Media.
25. Johnson DB. 1975 Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing* **4**, 77–84.
26. Consonni C, Laniado D, Montresor A. 2019 Discovering Topical Contexts from Links in Wikipedia. In *Wiki Workshop*.
27. Wulczyn E, Taraborelli D. 2017 Wikipedia Clickstream. .
28. Wikipedia contributors. 2019 Wikipedia:Manual of Style/Layout — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Wikipedia:Manual_of_Style/Layout&oldid=906190242#%22See_also%22_section. [Online; accessed 15-July-2019].
29. Nguyen P, Tomeo P, Di Noia T, Di Sciascio E. 2015 An evaluation of SimRank and Personalized PageRank to build a recommender system for the Web of Data. In *Proceedings of the 24th International Conference on World Wide Web* pp. 1477–1482.
30. Gupta P, Goel A, Lin J, Sharma A, Wang D, Zadeh R. 2013 Wtf: The who to follow service at twitter. In *Proceedings of the 22nd international conference on World Wide Web* pp. 505–514.