H-Index Manipulation by Merging Articles: Models, Theory, and Experiments

René van Bevern[*,†] and **Christian Komusiewicz** and **Rolf Niedermeier** and **Manuel Sorge**[†]
Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany,
{rene.vanbevern,christian.komusiewicz,rolf.niedermeier,manuel.sorge}
@tu-berlin.de

Toby Walsh[‡]
University of New South Wales and NICTA, Sydney, Australia, toby.walsh@nicta.com.au

Abstract

An author's profile on Google Scholar consists of indexed articles and associated data, such as the number of citations and the H-index. The author is allowed to merge articles, which may affect the H-index. We analyze the parameterized complexity of maximizing the H-index using article merges. Herein, to model realistic manipulation scenarios, we define a compatability graph whose edges correspond to plausible merges. Moreover, we consider multiple possible measures for computing the citation count of a merged article. For the measure used by Google Scholar, we give an algorithm that maximizes the H-index in linear time if the compatibility graph has constant-size connected components. In contrast, if we allow to merge arbitrary articles, then already increasing the H-index by one is NP-hard. Experiments on Google Scholar profiles of AI researchers show that the H-index can be manipulated substantially only by merging articles with highly dissimilar titles, which would be easy to discover.

1 Introduction

The H-index is a widely used measure for estimating the productivity and impact of researchers and research institutions. Hirsch [2005] defined the index as follows: a researcher has H-index h if h of the researcher's articles have at least h citations and all other articles have at most h citations. Several publicly accessible databases such as Web of Science, Scopus, ArnetMiner, and Google Scholar compute the H-index of researchers. Such metrics are therefore visible to hiring committees and funding agencies when comparing researchers and proposals.

Although the H-index of Google Scholar profiles is computed automatically, the owner of a profile can still manipulate her or his H-index by merging articles in their profile. The intention of this merging option is to identify different versions of the same article, for example a journal version and a version on arXiv.org, which are found as two different articles by Google's web crawlers. The merging of articles may change the H-index of a researcher since the merged article may have more citations than each of the original articles. This leaves the H-index of Google Scholar profiles vulnerable to manipulation by untruthful authors.

Increasing the H-index even by small values could be tempting, in particular for young researchers who are scrutinized more often than established researchers. For example, Hirsch [2005] estimates that, for the field of physics, the H-index of a successful researcher increases by roughly one per year of activity. Hence, an untruthful author might try to save years of research work with the push of a few buttons.

This type of manipulation has been studied by de Keijzer and Apt [2013]. In their model, each article in a profile comes with a number of citations. Merging two articles, one with x and one with y citations, replaces these articles by a new article with $x + y$ citations. This article may be then merged with further articles to obtain articles with even higher citation numbers. In this model, one can determine in polynomial time whether it is possible to improve the H-index by merging, but maximizing the H-index by merging is strongly NP-hard [de Keijzer and Apt, 2013]. We extend the results of de Keijzer and Apt [2013] in several ways.

1. We propose two further ways of measuring the number of citations of a merged article. One of them seems to be the measure actually used by Google Scholar.
2. We propose a model for restricting the set of allowed merge operations. Although Google Scholar allows merges between arbitrary articles, such a restriction is well motivated: an untruthful author may try to merge only similar articles in order to conceal the manipulation.
3. We consider the variant in which only a limited number of merges may be applied in order to achieve a desired H-index. This is again motivated by the fact that an untruthful author may try to conceal the manipulation by performing only few changes to her or his own profile.
4. We analyze all problem variants in the framework of parameterized complexity [Downey and Fellows, 2013; Flum and Grohe, 2006; Niedermeier, 2006]. This allows us, in some cases, to give efficient algorithms for realistic problem instances despite the NP-hardness of the problems in general.

[*] Now at Novosibirsk State University, Russian Federation.

[†] Supported by the DFG, project DAPA (NI 369/12).

[‡] Main work done during a visit at TU Berlin while supported by the Alexander von Humboldt Foundation, Bonn, Germany.

5. We evaluate our theoretical findings by performing experiments with real-world data based on the publication profiles of AI researchers.

Related work. A different way of manipulating the H-index is by strategic self-citations [Delgado López-Cózar *et al.*, 2014; Vinkler, 2013]; Bartneck and Kokkelmans [2011] consider approaches to detect these. Strategic self-citations take some effort and are irreversible. Thus, they can permanently damage an author's reputation. In comparison, article merging is easy, reversible and even justified in some cases.

Bodlaender and van Kreveld [2014] showed that in a previous version of the Google Scholar interface, it was NP-hard to decide whether a given set of articles can be merged at all.

A considerable body of work on manipulation can be found in the computational social choice literature [Faliszewski and Procaccia, 2010; Faliszewski *et al.*, 2010]. If we view citations as articles voting on other articles, then the problem we consider here is somewhat analogous to strategic candidacy [Dutta *et al.*, 2001].

1.1 Our models

We propose two new models for the merging of articles. These models take into consideration two aspects that are not captured by the model of de Keijzer and Apt [2013]:

1. The number of citations of an article resulting from a merge is not necessarily the sum of the merged articles. This is in particular the case for Google Scholar.
2. In order to hide manipulation, it would be desirable to only merge related articles instead of arbitrary ones. For example, one could only merge articles with similar titles.

To capture the second aspect, our model allows for constraints on the compatibility of articles. To capture the first aspect, we represent citations not by mere citation counts, but using a directed *citation graph* $D = (V, A)$. The vertices of D are the articles of our profile plus the articles that cite them, there is an arc (u, v) in D if article u cites article v.

Let $W \subseteq V$ denote the articles in our profile. In the following, these articles are called *atomic articles* and we aim to maximize our H-index by merging some articles in W. The result of a sequence of article merges is a partition $\mathcal{P}$ of W. We call each part $P \in \mathcal{P}$ with $|P| \geq 2$ a *merged article*. Note that having a merged article P corresponds to performing $|P| - 1$ successive merges on the articles contained in P. It is sometimes convenient to alternate between the partitioning and merging interpretations.

The aim is to find a partition $\mathcal{P}$ of W with a large H-index, where the *H-index of a partition* $\mathcal{P}$ is the largest number h such that there are at least h parts $P \in \mathcal{P}$ whose number $\mu(P)$ of citations is at least h. Herein, we have multiple possibilities of defining the measure $\mu(P)$ of citations of an article in $\mathcal{P}$. Before describing these possibilities, we introduce some notation.

Let $\deg^{\text{in}}_D(v)$ denote the indegree of an article v in the citation graph D, that is, its number of citations. Moreover, let $N^{\text{in}}_D(v) := \{u \mid (u, v) \in A\}$ denote the set of articles that cite v and $N^{\text{in}}_{D-W}(v) := \{u \mid (u, v) \in A \wedge u \notin W\}$ the set of articles that cite v and are not contained in W (thus, they may not be merged). For each part $P \in \mathcal{P}$, we consider the following three citation measures for defining the number $\mu(P)$ of citations of P. They are illustrated in Figure 1. The measure

$$\text{sumCite}(P) := \sum_{v \in P} \deg^{\text{in}}_D(v)$$

defines the number of citations of a merged article P to be the sum of the citations of the atomic articles it contains. This is the measure proposed by de Keijzer and Apt [2013]. In contrast, the measure

$$\text{unionCite}(P) := \left| \bigcup_{v \in P} N^{\text{in}}_D(v) \right|$$

defines the number of citations of a merged article P as the number of distinct atomic articles citing at least one atomic article in P. We verified empirically that, at the time of writing, Google Scholar used the unionCite measure. The measure

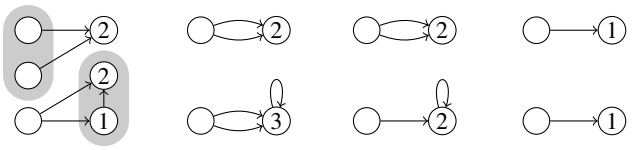
$$\text{fusionCite}(P) := \left| \bigcup_{v \in P} N^{\text{in}}_{D-W}(v) \right| + \sum_{P' \in \mathcal{P} \setminus \{P\}} \begin{cases} 1 & \text{if } \exists v \in P' \exists w \in P : (v, w) \in A, \\ 0 & \text{otherwise} \end{cases}$$

is, in our opinion, the most natural one: At most one citation of a part $P' \in \mathcal{P}$ to a part $P \in \mathcal{P}$ is counted. In contrast to the two other measures, merging two articles under the fusionCite measure may lower the number of citations of the resulting article and of other articles.

To model constraints on permitted article merges, we furthermore consider an undirected *compatibility graph* $G = (V, E)$. We call two articles *compatible* if they are adjacent in G. We say that a partition $\mathcal{P}$ of the articles W *complies* with G if for each part $P \in \mathcal{P}$ all articles in P are pairwise compatible, that is, if $G[P]$ is a clique. Thus, if the compatibility graph G is a clique, then there are no constraints: all partitions of W comply with G in this case.

Formally, for each measure $\mu \in \{\text{sumCite}, \text{unionCite}, \text{fusionCite}\}$, we are interested in the following problem:

H-INDEX MANIPULATION(μ)
Input: A citation graph $D = (V, A)$, a compatibility graph $G = (V, E)$, a set $W \subseteq V$ of articles, and a nonnegative integer h.



(a) Unmerged (b) sumCite (c) unionCite (d) fusionCite

Figure 1: Vertices represent articles, arrows represent citations, numbers are citation counts. The articles on a gray background in (a) have been merged in (b)–(d), and citation counts are given according to the measures sumCite, unionCite, and fusionCite, respectively. The arrows represent the citations counted by the corresponding measure.

Question: Is there a partition of W that complies with G and that has H-index at least h with respect to μ?

Throughout this work, we use $n := |V|$ to denote the number of input articles and $m := |E| + |A|$ to denote the overall number of arcs and edges in the two input graphs.

1.2 Our results

We study the complexity of H-INDEX MANIPULATION with respect to several structural features of the input instances. In particular, we consider the following three parameters:

- The size c of the largest connected component in the compatibility graph G. We expect this size to be small if only reasonable merges are allowed (or at least, if all merges have to appear reasonable).
- The number k of merges. An untruthful author would hide manipulations using a small number of merges.
- The H-index to be achieved. Although one is interested in maximizing the H-index, we expect this number also to be relatively small, since even experienced researchers seldom have an H-index of greater than 70.[1]

Table 1 summarizes our theoretical results. For example, we find that, with respect to the $\mathrm{unionCite}$ measure used by Google Scholar, it is easier to manipulate the H-index if only a small number of articles can be merged into one (small c). The $\mathrm{unionCite}$ measure is complex enough to make increasing the H-index by one an NP-hard problem even if the compatibility graph G is a clique. In contrast, for the $\mathrm{sumCite}$ measure and the compatibility graph being a clique, it can be decided in polynomial time whether the H-index can be increased by one [de Keijzer and Apt, 2013]. Due to space constraints, most proofs are omitted.[2]

We implemented the manipulation algorithms exploiting small k and small c. Experimental results show that all of our sample AI authors can increase their H-index by only three merges but that usually merging articles with highly dissimilar titles is required to obtain any improvement.

1.3 Preliminaries

We analyze H-INDEX MANIPULATION with respect to its classic and its parameterized complexity. The aim of parameterized complexity theory is to analyze problem difficulty not only in terms of the input size, but also with respect to an additional parameter, typically an integer p [Downey and Fellows, 2013; Flum and Grohe, 2006; Niedermeier, 2006]. Thus, formally, an instance of a parameterized problem is a pair (I, p) consisting of the input I and the parameter p. A parameterized problem with parameter p is *fixed-parameter tractable (FPT)* if there is an algorithm that decides an instance (I, p) in $f(p) \cdot |I|^{O(1)}$ time, where f is an arbitrary computable function depending only on p. Clearly, if the problem is NP-hard, we expect f to grow superpolynomially.

[1]The website `arnetminer.org` currently lists less than 90 researchers in computer science with H-index at least 70.

[2]A preliminary full version of the paper is available at http://arxiv.org/abs/1412.5498.

	$\mathrm{sumCite}$	$\mathrm{unionCite}$	$\mathrm{fusionCite}$
c	Solvable in $O(3^c c \cdot (n+m))$ time (Theorem 1)		NP-hard even for $c = 2$ (Theorem 2)
h	W[1]-hard but FPT if G is a clique (Corollary 2)		W[1]-hard (Corollary 1)
k	W[1]-hard (Theorem 4); FPT if G is a clique (Theorem 3)	W[1]-hard even if G is a clique (Theorem 5)	
	Improving H-index by one is NP-hard (Theorem 4), but poly-time if G is a clique [de Keijzer and Apt, 2013]	Improving H-index by one is NP-hard even if G is a clique (Theorem 6)	

Table 1: Summary of results for the citation measures $\mathrm{sumCite}$, $\mathrm{unionCite}$, $\mathrm{fusionCite}$, and the parameters "size c of the largest connected component of the compatibility graph G", "number k of allowed article merges", and "H-index h to achieve".

There are parameterized problems for which there is good evidence that they are not fixed-parameter tractable. Analogously to the concept of NP-hardness, the concept of W[1]-hardness was developed. It is widely assumed that a W[1]-hard problem cannot have a fixed-parameter algorithm. To show that a problem is W[1]-hard, a *parameterized reduction* from a known W[1]-hard problem can be used. This is a reduction that runs in $f(p) \cdot |I|^{O(1)}$ time and maps the parameter p to a new parameter p' that is bounded by some function $g(p)$.

The notion of a *problem kernel* tries to capture the existence of provably effective preprocessing rules [Guo and Niedermeier, 2007; Kratsch, 2014]. More precisely, we say that a parameterized problem has a problem kernel if every instance can be reduced in polynomial time to an equivalent instance whose size depends only on the parameter.

2 Compatibility graphs with small connected components

In this section, we analyze the parameterized complexity of H-INDEX MANIPULATION parameterized by the size c of the largest connected component of the compatibility graph. This parameterization is motivated by the fact that one would merge only similar articles and that usually each article is similar to only few other articles.

The following theorem shows that H-INDEX MANIPULATION is solvable in linear time for the citation measures $\mathrm{sumCite}$ and $\mathrm{unionCite}$ if c is constant. The algorithm exploits that, for these two measures, merging articles does not affect other articles. Thus, we can solve each connected component independently of the others.

Theorem 1. H-INDEX MANIPULATION(μ) *is solvable in* $O(3^c c \cdot (n+m))$ *time for* $\mu \in \{\mathrm{sumCite}, \mathrm{unionCite}\}$ *if the connected components of the compatibility graph* G *have size at most* c.

Proof. Clearly, articles from different connected components

of G cannot be together in a part of any partition complying with G. Thus, independently for each connected component C of G, we compute a partition of the articles of C that complies with G and has the maximum number of parts P with $\mu(P) \geq h$.

We first show that this approach is correct and then show how to execute it efficiently. Obviously, if an algorithm creates a partition $\mathcal{P}$ of the set W of our own articles that complies with G and has at least h parts P with $\mu(P) \geq h$, then we face a yes-instance. Conversely, if the input is a yes-instance, then there is a partition $\mathcal{P}$ of W complying with G and having at least h parts P with $\mu(P) \geq h$. Consider any connected component C of G and the restriction $\mathcal{P}_C = \{P \in \mathcal{P} \mid P \subseteq C\}$ of $\mathcal{P}$ to C. Note that each part in $\mathcal{P}$ is either contained in C or disjoint from it and, thus, $\mathcal{P}_C$ is a partition of C. Moreover, merging articles of one connected component does not affect the number of citations of articles in other connected components with respect to $\mathrm{sumCite}$ or $\mathrm{unionCite}$. Thus, if we replace the sets of $\mathcal{P}_C$ in $\mathcal{P}$ by a partition of C that has a maximum number of parts P with $\mu(P) \geq h$, then we obtain a partition that still has H-index at least h. Thus, our algorithm indeed finds a partition with H-index at least h.

We now show how to compute for each connected component C of G a partition that maximizes the number of parts with at least h citations. In order to achieve a running time of $O(3^c c \cdot (n+m))$, we employ dynamic programming. Let $V(C)$ denote the vertex set of C. First, for every connected component C of G and every $V' \subseteq V(C)$, we initialize a table

$$D[V'] := \begin{cases} 1 & \text{if } G[V'] \text{ is a clique and } \mu(V') \geq h, \\ 0 & \text{if } G[V'] \text{ is a clique and } \mu(V') < h, \\ -\infty & \text{otherwise.} \end{cases}$$

A table entry $D[V']$ thus stores whether merging V' results in an article with at least h citations. Obviously, if $G[V']$ is not a clique, then V' cannot be a part in any partition complying with G. Therefore, we set $D[V'] := -\infty$ in this case. All table entries $D[V']$ for all vertex subsets V' of all connected components of G can be computed in $O(2^c \cdot (n+m))$ time.

Now, for every vertex subset $V' \subseteq V(C)$ of a connected component C, we define $T[V']$ to be the maximum number of parts P with $\mu(P) \geq h$ in any partition of V'. Obviously,

$$T[V'] = \begin{cases} 0 & \text{if } V' = \emptyset, \\ \max_{V'' \subsetneq V'}(T[V''] + D[V' \setminus V'']) & \text{otherwise.} \end{cases}$$

After computing the table D, we can compute $T[V(C)]$ for each connected component C in $O(3^c c)$ time, since there are at most 3^c partitions of $V(C)$ into $V(C) \setminus (V' \cup V'')$, $V(C) \cap (V' \setminus V'')$ and $V(C) \cap V' \cap V''$. □

We have seen that H-INDEX MANIPULATION is solvable in linear time for the citation measures $\mathrm{sumCite}$ and $\mathrm{unionCite}$ if the compatibility graph has constant-size connected components. In contrast, constant-size components of the compatibility graph do not help when the $\mathrm{fusionCite}$ measure is used. This is shown by a reduction from the NP-hard 3-BOUNDED POSITIVE 1-IN-3-SAT problem [Denman and Foster, 2009].

Theorem 2. H-INDEX MANIPULATION($\mathrm{fusionCite}$) *is NP-hard even if*

i) the largest connected component of the compatibility graph has size two and

ii) the citation graph is acyclic.

Regarding (ii), note that citation graphs are often acyclic in practice as papers tend to cite only earlier papers. Thus, it is important that Theorem 2 does not require cycles in the citation graph.

3 Merging few articles or increasing the H-Index by one

In this section, we consider two variants of H-INDEX MANIPULATION: CAUTIOUS H-INDEX MANIPULATION, where we allow to merge at most k articles and H-INDEX IMPROVEMENT, where we ask whether it is possible to increase the H-index at all.

CAUTIOUS H-INDEX MANIPULATION is motivated by the fact that an untruthful author could try to conceal her or his tempering by merging only few articles. Formally, the problem is defined as follows, where $\mu \in \{\mathrm{sumCite}, \mathrm{unionCite}, \mathrm{fusionCite}\}$ as before.

CAUTIOUS H-INDEX MANIPULATION(μ)

Input: A citation graph $D = (V, A)$, a compatibility graph $G = (V, E)$, a set $W \subseteq V$ of articles, and non-negative integers h and k.

Question: Is there a partition $\mathcal{P}$ of W that

i) complies with G,

ii) has H-index at least h with respect to μ, and

iii) is such that the number $\sum_{P \in \mathcal{P}}(|P| - 1)$ of merges is at most k?

We show that CAUTIOUS H-INDEX MANIPULATION parameterized by k is fixed-parameter tractable only for the $\mathrm{sumCite}$ measure and when arbitrary merges are allowed, that is, the compatibility graph is a clique. Generalizing the compatibility graph or using more complex measures leads to W[1]-hardness with respect to k.

Since H-INDEX MANIPULATION is NP-hard, a natural question to ask, and an intuitively easier problem to solve, is whether the H-index can be improved at all. This variant was introduced by de Keijzer and Apt [2013]; it is defined as follows.

H-INDEX IMPROVEMENT(μ)

Input: A citation graph $D = (V, A)$, a compatibility graph $G = (V, E)$, and a set $W \subseteq V$ of articles.

Question: Is there a partition $\mathcal{P}$ of W that complies with G and has a larger H-index than $\mathcal{W}$ with respect to μ, where $\mathcal{W}$ is the singleton partition of W?

De Keijzer and Apt [2013] gave a polynomial-time algorithm for H-INDEX IMPROVEMENT($\mathrm{sumCite}$) if the compatibility graph is a clique. In contrast, we prove that generalizing the compatibility graph or using more complex measures leads to NP-hardness. We first give the tractable case of CAUTIOUS H-INDEX MANIPULATION and then turn to the hard cases.

Theorem 3. *If the compatibility graph G is a clique, then* CAUTIOUS H-INDEX MANIPULATION(sumCite) *is solvable in $O(9^k k^2 \cdot (n+m))$ time, where k is the number of allowed article merges.*

The result is based on a dynamic program, similar to the one in Theorem 1. If we generalize the compatibility graph, then we obtain the following hardness results by reductions from MULTICOLORED CLIQUE.

Theorem 4. *Parameterized by k,* CAUTIOUS H-INDEX MANIPULATION(sumCite) *is W[1]-hard.* H-INDEX IMPROVEMENT(sumCite) *is NP-hard.*

Now we restrict the compatibility graph to be a clique, and consider the measures unionCite and fusionCite. As mentioned above, we obtain hardness results; the (parameterized) reductions are from the INDEPENDENT SET problem.

INDEPENDENT SET
Input: An undirected graph H and a non-negative integer ℓ.
Question: Is there an *independent set* of size at least ℓ in H, that is, a set of ℓ pairwise nonadjacent vertices?

INDEPENDENT SET is known to be NP-hard and W[1]-hard with respect to ℓ [Downey and Fellows, 2013].

Theorem 5. *For $\mu \in \{\text{unionCite}, \text{fusionCite}\}$,* CAUTIOUS H-INDEX MANIPULATION(μ) *is W[1]-hard parameterized by k even if the compatibility graph is a clique.*

The reduction for Theorem 5 crucially relies on the fact that at most k merges are allowed. Hence, to show hardness for H-INDEX IMPROVEMENT, we need a different reduction.

Theorem 6. H-INDEX IMPROVEMENT(μ) *is NP-hard for $\mu \in \{$unionCite, fusionCite$\}$ even if the compatibility graph is a clique.*

Proof sketch. We give a polynomial-time reduction from INDEPENDENT SET. Let (H, ℓ) be an instance of INDEPENDENT SET and let $q := |E(H)|$. Without loss of generality, we assume that $q \geq \ell > 2$. We now construct an instance of H-INDEX IMPROVEMENT with citation graph D, a set V of articles, and a subset $W \subseteq V$ of own articles. The compatibility graph G will be a clique on all articles. We introduce citations so that the H-index of the singleton partition of W will be $q-1$, hence the goal in the constructed instance will be to achieve H-index at least q.

The article set W is partitioned into three parts $W = W_{\geq} \uplus W_{-1} \uplus W_{<}$. The first part, $W_{\geq}$, consists of $q-\ell-1$ articles, and for each article $w \in W_{\geq}$ we introduce q articles not in W that cite w and no other article. The second part, W_{-1}, consists of ℓ articles, and for each article $w \in W_{-1}$ we introduce $q-1$ articles not in W that cite w and no other article. The last part, $W_{<}$, contains the vertices of the INDEPENDENT SET instance, that is, $W_{<} := V(H)$. Finally, for each edge $\{u, v\} \in E(H)$ we introduce one article $e_{\{u,v\}}$ not in W that cites both u and v. This concludes the construction of the citation graph D. Note that the singleton partition of W has H-index $q-1$. Hence, we have created an instance (D, G, W) of H-INDEX IMPROVEMENT where we are looking to increase the H-index to at least q. Clearly, we can carry out this construction in polynomial time. Furthermore, since there are no self-citations, that is, no articles in W cite each other, for any subset P of W we have $\text{unionCite}(P) = \text{fusionCite}(P)$. We omit the proof of the equivalence of the two instances. □

4 Achieving a moderately large H-index

We now consider the H-index that we want to achieve as a parameter. This parameter is often not very large as researchers in the early stage of their career have an H-index below 20. Even for more experienced researchers the H-index seldom exceeds 70. Hence, in many cases the value of a desired H-index is sufficiently low to serve as useful parameter in terms of gaining efficient fixed-parameter algorithms.

We note that the reduction behind Theorem 4 also is a parameterized reduction to H-INDEX MANIPULATION with respect to the H-index we want to achieve. Hence, we have the following.

Corollary 1. H-INDEX MANIPULATION(sumCite) is W[1]-hard with respect to the H-index.

Note that the hardness also transfers to the unionCite and fusionCite measures. We now show that H-INDEX MANIPULATION(unionCite) is fixed-parameter tractable if the compatibility graph is a clique. Indeed, this result also holds for the sumCite measure. To this end, we describe a kernelization algorithm, that is, a polynomial-time data reduction algorithm that produces an equivalent instance whose size is bounded by some function of the parameter h. The first step is to simplify the instance by the following data reduction rule, which removes citations between articles in W.

Rule 1. If there is an article $w \in W$ such that the set $W' \subseteq W$ of articles cited by w is nonempty, then do the following. Add a new article v to $V \setminus W$, add citations from v to each article in W', and remove all outgoing citations from w.

It can be shown that it is safe to apply Rule 1. Let $W_{<}$ denote the set of articles that have less than h citations but at least one citation. The next step in our kernelization algorithm is to bound the number of articles that cite articles in $W_{<}$. To achieve this, we apply Algorithm 1, which greedily finds a solution if there are many articles that cite articles in $W_{<}$. Intuitively, it merges articles as long as merging makes some progress towards more articles with h citations.

Lemma 1. If there are at least $2h^2$ articles that cite articles in $W_{<}$, then Algorithm 1 finds a solution.

Thus, after applying Algorithm 1 we may assume that less than $2h^2$ articles cite articles in $W_{<}$. We now apply two further data reduction rules. The intuition behind the first rule is that if there is an article that cites a lot of articles in $W_{<}$, then many of those citations are irrelevant if the goal is to obtain H-index h. Thus, they can be safely removed.

Rule 2. If there is an article $v \in V$ that cites more than h^2 articles in $W_{<}$, then remove an arbitrary citation (v, w) outgoing from this article.

The next rule removes further unnecessary articles and citations from the instance. Its correctness is obvious.

Rule 3. If there is an article $w \in W$ that is not cited at all, then remove w from the instance. If there is an article $v \in V \setminus W$

Algorithm 1: Greedy Merge

Input: A citation graph $D = (V, A)$, a compatibility graph $G = (V, E)$, and a set of articles $W_< \subseteq V$, each with less than h and at least one citation.
Output: A partition $\mathcal{P}$ of $W_<$.

$\mathcal{P} \leftarrow \emptyset$
while $\exists a \in W_<$ **do**
  $B \leftarrow \{a\}$
  $W_< \leftarrow W_< \setminus \{a\}$
  while $(\text{unionCite}(B) < h) \wedge$
  $(\exists a \in W_< : \text{unionCite}(B \cup \{a\}) > \text{unionCite}(B))$
  do
    $B \leftarrow B \cup \{a\}$
    $W_< \leftarrow W_< \setminus \{a\}$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{B\}$
return $\mathcal{P}$

that does not cite any articles, then remove v from the instance. If there is an article in $W \setminus W_<$ that has more than h incoming citations, then remove one of these citations.

Applying first Rule 1 exhaustively, then Algorithm 1, and then Rules 2 and 3 exhaustively (if Algorithm 1 does not find a solution) results in a small instance.

Theorem 7. *If the compatibility graph is a clique, then an $O(h^4)$-article problem kernel for* H-INDEX MANIPULATION(μ) *with $\mu \in \{\text{sumCite}, \text{unionCite}\}$ is computable in polynomial time.*

While the problem kernel shown in Theorem 7 is rather large and its size certainly deserves improvement, it finally allows us to obtain the following classification result.

Corollary 2. If the compatibility graph is a clique, then H-INDEX MANIPULATION(μ) is fixed-parameter tractable with respect to the H-index for $\mu \in \{\text{sumCite}, \text{unionCite}\}$.

5 Experiments

To examine by how much authors can increase their H-indices when allowing only merges of articles with similar titles or when fixing the allowed number of merges, we implemented our algorithms for the parameter "maximum connected component size c of the compatibility graph" (Theorem 1) and for the parameter k of allowed merges (Theorem 3). We ran both algorithms using both the sumCite and unionCite measures. The algorithm for Theorem 3 does not necessarily compute the maximum possible H-index increase for unionCite (cf. Theorem 5), but we note that it yields a lower bound. Moreover, running it with sumCite yields an upper bound for the maximum achievable with unionCite.

Data acquisition. We crawled Google Scholar data of 22 selected authors of IJCAI'13. Our (biased) selection was based on capturing authors in their early career, for whom H-index manipulation seems most attractive. Specifically, we selected authors that have a Google Scholar profile, an H-index between 8 and 20, between 100 and 1000 citations, that are active between 5 and 10 years, and do not have a professor position.

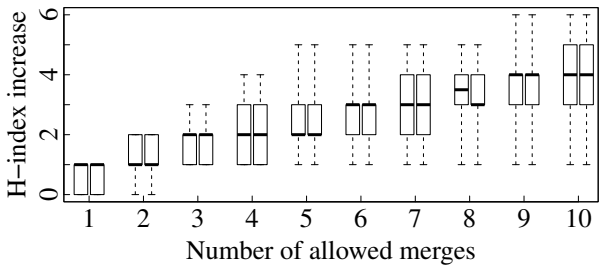


Figure 2: For each number k of allowed merges, the left box shows the H-index increase for sumCite, the right box shows lower bounds on the possible H-index increase for unionCite.

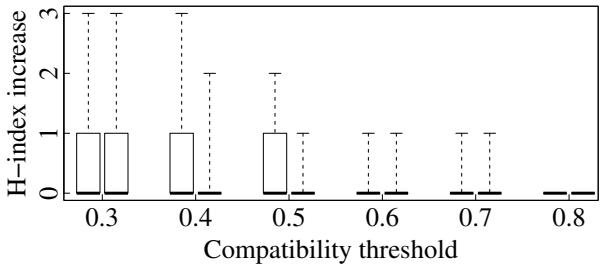


Figure 3: For each compatibility threshold t, the left box shows the H-index increase for sumCite, the right box for unionCite.

For each of the 22 authors, we computed upper and lower bounds for the H-index increase when allowing at most $k = 1, \ldots, 12$ merges and the maximum possible H-index increase when merging only articles whose titles have a similarity above a certain compatibility threshold $t = 0.1, 0.2, \ldots, 0.9$. The thresholding is described in more detail below.

Generating compatibility graphs. Compatibility graphs are constructed using the following simplified bag of words model: Compute for each article u the set of words $T(u)$ in its title. Draw an edge between articles u and v if $|T(u) \cap T(v)| \geq t \cdot |T(u) \cup T(v)|$, where $t \in [0, 1]$ is the *compatibility threshold*. For $t = 0$ the compatibility graph is a clique, for $t = 1$ only articles with the same title are adjacent. Inspection showed that for $t \leq 0.3$, already very dissimilar articles are considered compatible.

Experimental results. With a time limit of one hour on a 3.6 GHz Intel Xeon E5-1620 processor and a memory limit of 64 GB, our algorithms failed to solve many instances with a compatibility threshold $t \leq 0.2$ or allowing $k \geq 11$ merges. Instances with $k \leq 10$ and $t \geq 0.3$ were usually solved within few seconds and using at most 100 MB of memory. Thus, Figures 2 and 3 show results only for these instances.

Figure 2 shows the H-index increase over all authors for each number $k = 1, \ldots, 10$ of allowed article merges: the lower edge of a box is the 25th percentile and the upper edge is the 75th percentile, a thick bar is the median. The whiskers above and below each box extend to the maximum and minimum observed values. Remarkably, three merges

are sufficient for all of our sample authors to increase their H-index by at least one. To put the observed H-index increases in perspective, we measured that the unmanipulated H-index of our sample authors grows by 1.22 per year on average (which is higher than the one-per-year increase observed by Hirsch [2005] in physics). That is, from Figure 2, one can conclude that three merges can save almost 20 months of work for half of our sample authors.

Figure 3 shows the H-index increase over all authors for $\mathrm{unionCite}$ and each compatibility threshold $t = 0.3, 0.4, \ldots, 0.9$. Remarkably, when using a compatibility threshold $t \geq 0.4$, 75% of our sample authors cannot increase their H-index on Google Scholar. We conclude that increasing the H-index substantially by article merges should be easy to discover since it is necessary to merge articles with highly dissimilar titles for such a manipulation.

In similar experiments with 14 authors of IEEE Computer Society's *AI's 10 to Watch* 2011 and 2013 [AI's 10 to Watch, 2011; Zeng, 2013], more than half could increase their H-index using a single merge, but 75% could not increase their H-index with a compatibility threshold $t \geq 0.5$. Although the AI's 10 to Watch are more advanced in their career and have higher unmanipulated H-indices than the selected IJCAI authors, the results observed in both subject groups are similar.

6 Outlook

Clearly, it is interesting to consider merging articles in order to increase other measures than the H-index, like the g-index [Egghe, 2006; Woeginger, 2008a], the w-index [Woeginger, 2008b], or the $i10$-index of a certain author. The $i10$-index, the number of articles with at least ten citations, is also currently used by Google Scholar.

Furthermore, merging articles in order to increase one index might decrease other indices, like the overall number of citations. Hence, one could study the problem of increasing the H-index by merging without decreasing the overall number of citations or the $i10$-index below a predefined threshold.

Altogether, our experiments show that the merging option leaves some room for manipulation but that *substantial* manipulation requires merging visibly unrelated articles. Hiring committees that use the H-index in their evaluation thus should either examine the article merges more closely or rely on databases that do not allow article merges.

References

[AI's 10 to Watch, 2011] AI's 10 to Watch. *IEEE Intelligent Systems*, 26(1):5–15, 2011.

[Bartneck and Kokkelmans, 2011] Christoph Bartneck and Servaas Kokkelmans. Detecting h-index manipulation through self-citation analysis. *Scientometrics*, 87(1):85–98, 2011.

[Bodlaender and van Kreveld, 2014] Hans L. Bodlaender and Marc van Kreveld. Google Scholar makes it hard—the complexity of organizing one's publications. *CoRR*, abs/1410.3820, 2014.

[Delgado López-Cózar *et al.*, 2014] Emilio Delgado López-Cózar, Nicolás Robinson-García, and Daniel Torres-Salinas. The Google Scholar experiment: How to index false papers and manipulate bibliometric indicators. *Journal of the Association for Information Science and Technology*, 65(3):446–454, 2014.

[Denman and Foster, 2009] Richard Denman and Stephen Foster. Using clausal graphs to determine the computational complexity of k-bounded positive one-in-three SAT. *Discrete Applied Mathematics*, 157(7):1655–1659, 2009.

[Downey and Fellows, 2013] Rod G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

[Dutta *et al.*, 2001] Bhaskar Dutta, Matthew O. Jackson, and Michel Le Breton. Strategic candidacy and voting procedures. *Econometrica*, 69(4):1013–1037, 2001.

[Egghe, 2006] Leo Egghe. Theory and practise of the g-index. *Scientometrics*, 69(1):131–152, 2006.

[Faliszewski and Procaccia, 2010] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[Faliszewski *et al.*, 2010] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[Flum and Grohe, 2006] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[Guo and Niedermeier, 2007] Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007.

[Hirsch, 2005] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46):16569–16572, 2005.

[de Keijzer and Apt, 2013] Bart de Keijzer and Krzysztof R. Apt. The H-index can be easily manipulated. *Bulletin of the EATCS*, 110:79–85, 2013.

[Kratsch, 2014] Stefan Kratsch. Recent developments in kernelization: A survey. *Bulletin of the EATCS*, 113:58–97, 2014.

[Niedermeier, 2006] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Number 31 in Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.

[Vinkler, 2013] Péter Vinkler. Would it be possible to increase the Hirsch-index, π-index or CDS-index by increasing the number of publications or citations only by unity? *Journal of Informetrics*, 7(1):72–83, 2013.

[Woeginger, 2008a] Gerhard J. Woeginger. An axiomatic analysis of Egghe's g-index. *Journal of Informetrics*, 2(4):364–368, 2008.

[Woeginger, 2008b] Gerhard J. Woeginger. An axiomatic characterization of the Hirsch-index. *Mathematical Social Sciences*, 56(2):224–232, 2008.

[Zeng, 2013] Daniel Zeng. AI's 10 to watch. *IEEE Intelligent Systems*, 28(3):86–96, 2013.