Fakultät für Elektrotechnik und Informatik

Fachgebiet Algorithmik und Komplexitätstheorie

Technische Universität Berlin

g-Indexmanipulation: Algorithmen zum Auflösen von Artikelvereinigungen

Bachelorarbeit Informatik

Von: Amin Abid
Matrikelnummer: 3471217

Erstgutachter: Prof. Dr. Rolf Niedermeier
Zweitgutachter: Prof. Dr. Uwe Nestmann

Betreuer:
Hendrik Molter
Manuel Sorge

15. Mai 2017

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig, sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, 15. Mai 2017

.....

Abstract

A bibliometric indicator measures the productivity of a researcher based on the citation count of his articles. Two established of these indicators are the h-Index and the g-Index. Such bibliometric indicators are calculated by public accessible article databases like Aminer, Google Scholar, Scopus and Web of Science. The h-Index describes the biggest count h of articles by the author which are at least h times cited and the g-Index describes the biggest count g of articles that are cited at least g times on average. Google Scholar permits the authors to merge multiple articles to one to signify different version of the same article. In this work we address the possibility of manipulating the g-Index by extracting single articles from these merged articles. Concerning Google Scholars method of counting citations of merged articles it is already known that the problem of achieving a certain h-Index by doing such extractions is possible in linear time. We reveal that this problem is NP-hard regarding the g-Index. Furthermore we develope an exponential time algorithm that solves it on which we do experiments. In these experiments we discern that the required runtime is relatively small and that improvements can generally only be achieved to a certain extend.

Zusammenfassung

Ein bibliometrischer Indikator bewertet daran, wie oft ein Forscher zitiert wurde, seine Leistung. Zwei etablierte dieser Indikatoren sind der h-Index und der g-Index. Solche bibliometrischen Indikatoren werden in öffentlich zugänglichen Artikel-Datenbanken wie AMiner, Google Scholar, Scopus, und Web of Science berechnet. Der h-Index beschreibt die größte Anzahl h an Artikeln eines Autors, die jeweils mindestens h mal zitiert werden und der g Index die größte Anzahl g an Artikeln, die im Durchschnitt mindestens g mal zitiert werden. Google Scholar erlaubt es den Autoren durch das Zusammenfügen ihrer Artikel, verschiedene Versionen des selben Artikels zu kennzeichnen. Wir widmen uns in dieser Arbeit der Frage, inwieweit man, durch Extraktionen einzelner Artikel aus diesen zusammengefügten Artikeln, den g-Index manipulieren kann. Es ist bereits bekannt, dass für Google Scholar's Zählweise der Zitate zusammengefügter Artikel, das Erreichen eines bestimmten h-Indexes durch Extraktionen in linearer Laufzeit möglich ist. Wir zeigen, dass dieses Problem für den g-Index hingegen NP-schwer ist. Des weiteren entwickeln wir hierfür einen Exponentialzeit-Algorithmus, auf dem wir anschließend Experimente durchführen. In diesen kommen wir zur Erkenntnis, dass die benötigte Laufzeit sich in der Praxis als relativ gering herausstellt und im allgemeinen nur bedingt eine Verbesserung des g-Indexes erzielt werden kann.

Inhaltsverzeichnis

1 Einleitung

Eine der üblichsten Methoden die Qualität, die Produktivität und den Einfluss von Forschern, Institutionen, etc. zu bewerten ist die Anzahl der Zitate ihrer verfassten Artikel zu betrachten. Für diesen Zweck existieren mehrere, so genannte bibliometrische Indikatoren, die die Zitate der verfassten Artikel eines Autors unterschiedlich evaluieren. Besonders beliebt hierfür ist der so genannte h-Index, der die größte Anzahl h von Artikeln eines Autors beschreibt, die mindestens h mal zitiert werden. Dieser wurde 2005 von Jorge E. Hirsch[8] eingeführt. Eine besonders simple alternative für den h-Index stellt der i10-Index dar, der 2011 als einer der verwendeten Indizes in Google Scholar eingeführt wurde. Er beschreibt die größte Anzahl an Artikeln mit mindestens 10 Zitaten [2]. Eine andere Alternative zum h-Index ist der 2006 von Leo Egghe[6] entwickelte g-Index, der die größte Menge g an Artikeln eines Autors beschreibt, die im Durchschnitt g Zitationen aufweisen. Damit unterscheidet er sich vom h-Index dadurch, dass alle Zitate der g ausgewählten Artikel in die Wertung eingehen, während es beim h-Index keine Rolle spielt, wie viele Zitate die Artikel im einzelnen besitzen, solange sie h übersteigen. Damit ist der g-Index mindestens so groß wie der h-Index und potentiell größer als dieser, wenn die mehr als g mal zitierten Artikel für die weniger als g zitierten Artikel im Durchschnitt kompensieren können.

Diese bibliometrische Indikatoren werden von öffentlich zugänglichen Artikel-Datenbanken, wie AMiner, Google Scholar, Scopus, und Web of Science berechnet und sind damit für jeden sichtbar. Dementsprechend werden diese von Personaleinstellungs- und Finanzierungs-Ausschüssen herangezogen, um mithilfe dieser die Leistung von Forschern zu vergleichen. Damit ist es im Interesse von Autoren diese Indikatoren so weit wie möglich zu erhöhen, um ihre Karriere voranzutreiben. Diese Erhöhung sollte normalerweise, im Sinne solcher Indikatoren, durch das Verfassen von relevanten Artikeln geschehen, allerdings bestehen einige Methoden diese anderweitig zu manipulieren. Eine dieser resultiert daraus, dass Autoren die Möglichkeit besitzen mehrere ihrer Artikel in den besagten Datenbanken zu einem einzigen Artikel zusammenzufügen. Dies soll es ihnen erlauben verschiedene Versionen des selben Artikels zu kennzeichnen. Beispielsweise würde ein Forscher eine Version eines Artikels aus einer Wissenschaftlichen Zeitschrift mit der entsprechenden Version von arXiv.org zusammenfügen, um sein Profil übersichtlich zu halten. Das Problem, dass hierbei besteht ist aber, dass Autoren zusammengefügte Artikel prinzipiell frei bilden und auch wieder auftrennen können, um ihre Indizes zu manipulieren. Ihnen wird nämlich ein großes Maß an Freiheit bei der Verwaltung dieser gelassen. Es gibt keine Regelung dafür welche Artikel zusammengefügt werden müssen und welche nicht. Autoren werden damit lediglich durch ihre eigene Moral und das Risiko einer möglichen Rufschädigung davon abgehalten, diese Freiheit beim Zusammengefügen ihrer Artikel zum Zweck der Verbesserung ihrer Indizes zu missbrauchen. Besonders verlockend wäre eine solche Manipulation für Forscher, die noch relativ früh in ihrer Karriere sind und dementsprechend strenger nach ihrer Fachkompetenz hinterfragt werden. Voraussetzung ist es natürlich, dass eine solche Manipulation auch praktikabel umsetzbar ist.

1.1 Literaturübersicht

Es gibt diverse Arbeiten, die sich mit der Frage befasst haben, inwieweit diese Möglichkeit zum zusammenfügen und aufteilen von Artikeln das Risiko der Manipulation der Indizes birgt. De Keijzer und Apt [5] befassten sich 2013 mit der potentiellen Manipulation des h-Indexes durch das Zusammengefügen von Artikel. Dabei zeigten sie, dass es in Polynomial-Zeit möglich ist, eine beliebige Verbesserung des h-Indexes zu erreichen. Durch das Zusammenfügen der Artikel einen bestimmten h-Index zu erreichen, oder den h-Index zu maximieren stellte sich jedoch als NP-schwer heraus.

Bevern et al. [3] untersuchten 2016 ebenfalls die Komplexität der Maximierung des h-Indexes durch das zusammenfügen von Artikeln. Sie betrachten dabei unter anderem zusätzlich verschiedene Zitationsmetriken für die Berechnung der Zitate eines zusammengefügten Artikels. Außerdem untersuchten sie die Einschränkung, nur Artikel mit ähnlichen Titeln zusammenzufügen, um die Manipulation zu verbergen. Die Ähnlichkeiten der verschiedenen Titel wurden durch Kanten in einen ungerichteten Graphen modelliert - dem Ähnlichkeitsgraph. Sie kamen für die Zitationsmetrik, welche von Google Scholar benutzt wird und die wir in dieser Arbeit auch ausschließlich behandeln werden, zum Ergebnis, dass das h-Index-Maximierungsproblem für konstant-große größte Zusammenhangskomponente des Ähnlichkeitsgraphen auch in Linear-Zeit lösbar ist. Ansonsten ist es exponentiell bezüglich dessen Größe. Die besagte Zitationsmetrik von Google Scholar wurde hier als Unioncite bezeichnet, was wir auch tun werden.

Im selben Jahr analysierten Elkind und Pavlou [9] darauf die Beeinflussung durch das Zusammenfügen von Artikeln für den g- und i10-Index und betrachteten dabei ebenfalls die Einschränkungen durch einen Ähnlichkeitsgraphen und den gleichen Zitationsmetriken, wie auch schon zuvor Bevern et al. [3]. Sie kamen zur Erkenntnis, dass der g- und der i10-Index durch das zusammenfügen von Artikeln im Allgemeinen leichter zu manipulieren sind, als der h-Index. Für die g-Index-Manipulation bezüglich Unioncite ist dieses Problem jedoch dennoch NP-schwer. Dies gilt sowohl für das erreichen eines bestimmten g-Indexes, als auch für eine beliebige Verbesserung dessen.

Eine Nachfolgearbeit der erwähnten Arbeit von Bevern et al. [4] befasst sich hingegen, im Gegensatz zur Vorgängerarbeit, die das Zusammenfügen betrachtete, mit dem *Aufbrechen* bereits bestehender zusammengefügter Artikel, um den h-Index zu verbessern. Dabei wurden unter anderem die *Extraktionen einzelner Artikel* aus den zusammengefügten Artikeln betrachten, um einen bestimmen h-Index zu erreichen. Speziell für Unioncite war dies in Linear-Zeit lösbar.

Wir werden uns stark an den Modellierungs-Konzepten der beiden Arbeiten von Bevern et al. orientieren und uns spezifisch ebenfalls damit befassen, inwieweit die Manipulation des g-Indexes durch Extraktionen *einzelner Artikel* aus ihren zusammengefügten Artikeln praktikabel ist. Als Zitationsmetrik betrachten wir dabei wie bereits erwähnt ausschließlich Unioncite, welches von Google Scholar verwendet wird und auch schon in diesen beiden Arbeiten definiert wurde.

1.2 Beitrag und Struktur der Arbeit

In dieser Arbeit untersuchen wir also die Komplexität des Problems, welche einzelnen Artikel man aus den zusammengefügten Artikeln eines Autors extrahieren muss, um einen bestimmten g-Index zu erreichen.

- Hierfür untersuchen und formalisieren wir in Abschnitt 3 die Eigenschaften und den Einfluss von Extraktionen auf die Zitation der Artikel des Autors.
- Wir kommen zur Erkenntnis, dass unser g-Index-Extraktions-Problem im Gegensatz zu seiner in Linearzeit lösbaren h-Index-Version NP-schwer ist, was wir auch in Abschnitt 4 durch einen Reduktionsbeweis zeigen.
- Anschließend beschreiben wir in Abschnitt 5 einen Algorithmus, der das Problem in exponentieller Laufzeit im Verhältnis zur Eingabegröße löst. Der exponentielle Teil der Komplexität des Algorithmus wird jedoch durch die Größe M_{max} des größten zusammengefügten Artikels limitiert. Für alle konstanten M_{max} hat der Algorithmus die gleiche polynomielle Laufzeit.
- Wir machen darauffolgend in Abschnitt 6 ein realitätsnahes Experiment, bei dem unsere Implementierung des Algorithmus für alle Testinstanzen innerhalb von Sekunden beendet, da die M_{max} in der Praxis gering sind.

Unseren Ergebnissen nach stellt der Rechenaufwand demnach in der Praxis keine Hürde bezüglich einer Manipulation des g-Indexes durch Artikelextraktionen dar. Bei einer Hand von Autoren unserer Testdaten kam es außerdem zu einer g-Index-Verbesserungen.

2 Notation und Definitionen

In diesem Abschnitt werden wir den g-Index kurz mit dem h-Index vergleichen und den g-Index an einem Beispiel genauer betrachten. Bevor wir dies tun können, müssen wir uns zunächst einige Grundbegriffe aneignen und genauer definieren.

2.1 Zitationsgraph und Extraktionen

Definition (Zitationsgraph)**.** *Ein Zitationsgraph ist ein gerichteter Graph $D = (V, Z)$, der aus den Knoten V und den Kanten Z besteht. Die Knotenmenge V stellt dabei die Menge aller existierenden einzelnen Artikel dar. Eine gerichtete Kante von einem Knoten u zu einem Knoten v existiert genau dann, wenn Artikel u Artikel v zitiert [3].*

Die einzelnen, nicht zusammengefügten Artikel, die in V enthalten sind, bezeichnen wir als *atomare Artikel*. Die Menge von atomaren Artikeln des betrachteten Autors wird durch eine Menge $W \subseteq V$ verkörpert. Die atomaren Artikel aus W können dann durch den Autor *zusammengefügt* werden. Welche atomaren Artikel des Autors mit welchen anderen zusammengefügt

werden, wird durch eine *Partition* $\mathcal{P}$ von W modelliert. Dabei werden Artikel $M \in \mathcal{P}$ mit $|M| \geq 2$ in unseren Haupt-Referenzarbeiten als zusammengefügte Artikel bezeichnet [3, 4]. Wir werden aber der sprachlichen Einfachheit halber auch Artikel $M \in \mathcal{P}$ mit nur einem atomaren Artikel als solche bezeichnen. Die Begriffe „Artikel“ und „zusammengefügter Artikel“ benutzen wir im Folgendem gleichwertig für die Elemente aus $\mathcal{P}$ und verwenden Zweiteren hauptsächlich um zu betonen, dass es sich nicht um einen atomaren Artikel handelt.

Wir bezeichnen im weiteren Verlauf dieser Arbeit außerdem $M_{v,\mathcal{P}}$ als den zusammengefügten Artikel A, der in der Partition $\mathcal{P}$ den atomaren Artikel v enthält, also

$$M_{v,\mathcal{P}} := \{u \mid A \in \mathcal{P} \wedge v \in A \wedge u \in A\}.$$

Zu beachten ist hierbei, dass jedes $v \in W$, entsprechend der Eigenschaften einer Partition, in *genau* einem zusammengefügten Artikel enthalten ist.

Definition (Extraktionen von atomaren Artikeln)**.** *Die Extraktion eines atomaren Artikels v aus seinem zusammengefügte Artikel $M_{v,\mathcal{P}}$, bezüglich einer Partition $\mathcal{P}$ einer Menge atomarer Artikel W, beschreiben wir durch die Partition $\mathcal{P}_v$ die man erhält, wenn man v aus $M_{v,\mathcal{P}}$ extrahiert. In Formeln,*

$$\mathcal{P}_v := (\mathcal{P} \setminus \{M_{v,\mathcal{P}}\}) \cup \{\{v\}\} \cup \{M_{v,\mathcal{P}} \setminus \{v\}\} = \bigcup_{A \in \mathcal{P}} \{A \setminus \{v\}\} \cup \{\{v\}\}.$$

Extraktionen einer Folge von k atomaren Artikeln f_i mit $i \in \{1, \ldots, k\}$ geben wir mithilfe von Tupeln aus diesen an. Wir definieren die Partitionen, die man dabei erhält als

$$\mathcal{P}_{(f_1,\ldots,f_k)} := \bigcup_{M \in \mathcal{P}} \{M \setminus \{f_1, \ldots, f_k\}\} \cup \{\{f_1\}, \ldots, \{f_k\}\}.$$

Die Menge der validen Extraktionsfolgen legen wir fest als

$$\mathcal{F} := \{(f_1, f_2, \ldots, f_k) \mid f_1, f_2, \ldots, f_k \in W \wedge \forall i, j \in \{1, 2, \ldots, k\}. i \neq j \rightarrow f_i \neq f_j\}.$$

Es muss also gelten, dass alle zu extrahierenden Artikel verschiedene atomare Artikel des Autors sind.

2.2 Zitationsmetriken und bibliometrische Indikatoren

Es stellt sich nun die Frage wie man bewertet, wie oft ein zusammengefügter Artikel zitiert wurde. Dies wird durch eine *Zitationsmetrik* bestimmt. Hierbei handelt es sich um eine Funktion $\mu(M)$, die einem Artikel M aus der Partition $\mathcal{P}$, unter Berücksichtigung der eingehenden Kanten der enthaltenen atomaren Artikel, eine natürliche Zahl zuordnet. Diese zugeordneten Zahlen werden wir als *Zitationswertungen* bezeichnen. Neben *Unioncite*, existieren als Zitationsmetriken auch *Sumcite* und *Fusioncite* [3]. *Die Zitationsmetrik auf die wir hauptsächlich eingehen werden ist Unioncite, die auch in Google Scholar verwendet wird. Wenn wir also die Funktion μ erwähnen, ist immer von Unioncite die Rede.*

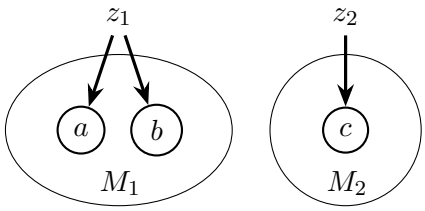


Abbildung 1: Die dargestellten Artikel M_1 und M_2 haben beide eine Zitationswertung von eins, da zitierende atomare Artikel die in einem zusammengefügten Artikel mehrmals zitieren nur einmal gezählt werden.

Definition (Unioncite)**.** *Unioncite ist bezüglich eines Zitationsgraphs $D = (V, Z)$, für einen Artikel $M \in \mathcal{P}$, als Anzahl verschiedener atomarer Artikel, die einen atomaren Artikel in M zitieren definiert, also*

$$\mu(M) :=\mid \bigcup_{v \in M} N_D^{in}(v) \mid .$$

Für eine ganze Partition $\mathcal{P}$ von W definieren wir μ als die Summe der Zitationswertungen aller Artikel M in dieser. In Formelschreibweise ist dies demnach

$$\mu(\mathcal{P}) := \sum_{M \in \mathcal{P}} \mu(M).$$

Als Zitationsindex, der die Leistung des Autors, an den Zitationen seiner Artikel bewerten soll, gibt es wie zuvor erwähnt unter anderem den *h-Index*, sowie den *g-Index*[8][6].

Definition (h-Index)**.** *Der h-Index einer Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$ ist definiert, als die größte Zahl h, sodass gilt*

$$h \leq |\{M | M \in \mathcal{P} \wedge \mu(M) \geq h\}|.$$

Wir konzentrieren uns jedoch in dieser Arbeit auf den *g-Index*.

Definition (g-Index)**.** *Der g-Index einer Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$ ist definiert, als die größte Zahl g, sodass es g Artikel $M_1, \ldots, M_g \in \mathcal{P}$ gibt, für die gilt*

$$g^2 \leq \sum_{i \leq g} \mu(M_i).$$

Für beide Indizes sind lediglich eine bestimmte Anzahl x der am meisten zitierten Artikel relevant. Diese beschreiben wir mit folgender Hilfsdefinition *top*, die die x Artikel, mit der größten Zitationswertung in der zu betrachtenden Partition $\mathcal{P}$ darstellt:

$$\text{top}(\mathcal{P}, x) := \arg\max_{T \subseteq \mathcal{P}} \{\mu(T) \mid |T| \leq x\}.$$

Zu beachten ist jedoch, dass *top* keine wohldefinierte Funktion ist. Dies ist der Fall, wenn mehrere Artikel die selbe Zitationswertung besitzen und für den Artikel mit der kleinsten Zitationswertung der ausgewählten Artikel in Frage kommen. Um dies entgegenzuwirken gehen wir davon

aus, dass wir stets eine bestimmte feste Auswahl von Artikeln wählen, die $\text{top}(\mathcal{P}, x)$ erfüllt. Dies lässt sich als eine bestimmte Priorisierungs-Rangfolge der Artikel, bei gleichen Zitationswertungen vorstellen. Wenn wir also behaupten, dass ein Artikel A ein Element in $\text{top}(\mathcal{P}, x)$ ist, dann bedeutet das, dass genau solch eine Auswahl für $\text{top}(\mathcal{P}, x)$ existiert, von der wir auch ausgehen.

topGet ist eine weitere Hilfsdefinition, die uns die Möglichkeit gibt, den Artikel mit der i-kleinsten Zitationswertung in den Artikeln mit den x größten Zitationswertungen in $\mathcal{P}$ zu beschreiben. Dabei sollen im Falle, dass $|\mathcal{P}| < x$ ist, zusätzlich $x - |\mathcal{P}|$ „leere“ Artikel vorgestellt werden, die eine Zitationswertung von null haben:

$$\text{topGet}(\mathcal{P}, x, i) := \underset{v \in \text{top}(\mathcal{P},x) \setminus \text{top}(\mathcal{P},x-i)}{\arg\max} \{\mu(v)\}.$$

Erneut besteht hier keine Wohldefiniertheit. Wir gehen wieder davon aus, dass eine Priorisierungs-Rangfolge der Artikel besteht, die bei gleichen Zitationswertungen mehrerer Artikel eine feste Auswahl trifft. Wenn wir also behaupten ein Artikel A ist gleich $\text{topGet}(\mathcal{P}, x, i)$, existiert eine Priorisierungs-Reihenfolge, sodass dies erfüllt wird und für die die $\text{topGet}(\mathcal{P}, x, i)$ für alle i verschiedene Artikel ausgeben. Von dieser einen gehen wir dann auch aus.

Das genaue Problem, dass wir mit unserem Algorithmus letztendlich lösen wollen, wird sich auf ein bestimmtes g konzentrieren. Es lässt sich wie folgt beschreiben.

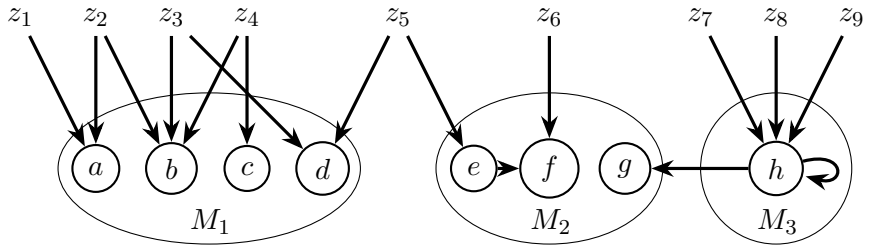
g-Index-Manipulation durch Artikel-Extraktion
Eingabe: Ein Zitationsgraph $D = (V, Z)$, eine Menge $W \subseteq V$ von Artikeln des Autors, eine Partition $\mathcal{P}$ von W und eine natürliche Zahl g.
Ausgabe: Eine Folge F, sodass $\mu(\text{top}(\mathcal{P}_F, g)) \geq g^2$ oder $\perp$, falls keine solche Folge existiert.

Beim h-Index müssen die *Zitationswertungen der Artikel in* $\text{top}(\mathcal{P}, h)$ also jeweils über dem Schwellenwert h liegen. Wie weit sie über diesen Wert liegen spielt dabei keine Rolle. Der Algorithmus, um nach einem h-Index zu prüfen, besteht deshalb ganz einfach aus Extraktion der atomaren Artikel, die eine Zitationswertung größer gleich h besitzen [4]. So einfach ist es jedoch nicht für den g-Index. Im Gegensatz zum h-Index kommt es für ihn auf jeden einzelnen Zitationspunkt der Artikel in $\text{top}(\mathcal{P}, g)$ an. Wir werden später sehen, dass bei einer Extraktion sowohl der extrahierte atomare Artikel, als auch der Restartikel potentiell eine kleinere Zitationswertung aufweisen als der Ursprungsartikel und maximal genau die gleiche besitzen. Deshalb ist es zum einen von Bedeutung, dass beide Teilartikel auch nach der letzten Extraktion in den g Artikeln mit den größten Zitationswertungen enthalten bleiben, da sonst unnütz Zitationen aus diesen verloren gehen. Zum anderen muss auch die Summe der Zitationswertungen durch die Extraktion größer werden, da sonst kein Fortschritt erlangt werden kann. Es sei vorweg zu erwähnen, dass unser Algorithmus das Ziel hat, die Extraktionsfolge zu finden, die die maximal erreichbare Erhöhung der Summe an Zitationswertungen in den Top g erzielt, anstatt dass wir eine beliebige suchen, die die Bedingung erfüllt. Wie wir später beweisen werden, ist das Entscheidungsproblem, ob überhaupt eine Folge existiert, die den gewünschten g-Index erreicht bereits NP-schwer.

$$V = \{a, b, c, d, e, f, g, h, z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9\}$$

$$Z = \{(z_1, a), (z_2, a), (z_2, b), (z_3, b), (z_3, d), (z_4, b), (z_4, c), (z_5, d),$$
$$(z_5, e), (z_6, f), (z_7, h), (z_8, h), (z_9, h), (h, g), (e, f), (h, h)\}$$

$$W = \{a, b, c, d, e, f, g, h\}\ \mathcal{P} = \{\{a, b, c, d\}, \{e, f, g\}, \{h\}\}$$



	a	b	c	d	e	f	g	h	M_1	M_2	M_3	$\mathcal{P}$
μ	2	3	1	2	1	2	1	4	5	4	4	13

	a	b	c	d	e	f	g	h
Gain in $\mathcal{P}$	1	3	1	1	0	0	0	0

Abbildung 2: Beispiel-Zitationsgraph

Schauen wir uns nun einen Beispiel-Zitationsgraph an (Abbildung 2). Wir bezeichnen hierbei zitierende atomare Artikel, die nicht in W enthalten sind mit einem z_i. Die M_i stellen zusammengefügte Artikel dar. Der h-Index beträgt für dieses Beispiel drei, da drei zusammengefügte Artikel $M_i \in \mathcal{P}$ existieren, so dass für alle diese gilt $\mu(M_i) \geq 3$. Auch der g-Index ist drei da $(\mu(M_1) + \mu(M_2) + \mu(M_3)) \geq 3^2$. Es ist zu beobachten, dass auch Zitationen von atomaren Artikeln des Autors in die Zitationswertungen gehen (h zitiert g). Dies gilt gleichermaßen für Selbstzitate (h) und Zitationen durch andere atomare Artikel im selben zusammengefügten Artikel (e nach f).

Ebenfalls ist hierbei zu beobachten, dass es bei Unioncite nicht in die Wertung mit eingeht, wie oft ein atomarer Artikel in einem zusammengefügten Artikel M zitiert. Jeder zitierende atomare Artikel wird lediglich einmal in der Zitationswertung $\mu(M)$ mit einer Erhöhung um eins einberechnet, wenn mindestens einmal in M zitiert wird (siehe hierfür auch Abbildung 1). Dies hat zur Folge, dass falls ein atomarer Artikel v aus $M_{v,\mathcal{P}}$ extrahiert wird, $\mu(\mathcal{P}_v)$ im Vergleich zu $\mu(\mathcal{P})$ um die Anzahl der zitierenden atomaren Artikel erhöht wird, die gleichzeitig v und andere atomare Artikel in $M_{v,\mathcal{P}}$ zitieren. In der entstehenden Partition $\mathcal{P}_v$ zitieren diese nun aber sowohl $\{v\}$, als auch den Restartikel $M_{v,\mathcal{P}} \setminus \{v\}$ und damit einen Artikel mehr. Die Summe der Zitationswertungen der Artikel des Autors wurde damit künstlich erhöht. Um die dadurch

mögliche Manipulation des g-Index formalisieren zu können, führen wir die für diese Arbeit grundlegenden Begriffe des *Gain* und *Realgain* ein.

3 Eigenschaften von Extraktionen

In diesem Abschnitt werden wir durch die Begriffe des *Gain* und des *Realgain* die Auswirkungen von einzelnen Extraktionen eines atomaren Artikels auf die Zitationswertungen eines Autors genauer formalisieren und deren Eigenschaften beschreiben. Der Gain beschreibt dabei den Anstieg der Zitationen durch die Extraktion insgesamt, wobei der Realgain nur die Zitationen in den g Artikeln mit den größten Zitationswertungen betrachtet.

3.1 Gain

Definition (Gain)**.** *Der Gain eines atomaren Artikels v und einer Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$ beschreibt den Zitationsanstieg, der durch die Extraktion von v aus $M_{v,\mathcal{P}}$ in der entstehenden Partition $\mathcal{P}_v$ im Vergleich zur alten $\mathcal{P}$ erzielt werden kann, also*

$$\text{gain}(\mathcal{P}, v) := \mu(\mathcal{P}_v) - \mu(\mathcal{P}).$$

Der *Gain* lässt sich wie bereits erwähnt aus der Anzahl der zitierenden Artikel bestimmen, die sowohl den zu extrahierenden Artikel, als auch den Restartikel zitieren, was durch einsetzen der bisher eingeführten Definitionen im nun folgenden Lemma 1b aufzeigt wird. Lemma 1a ermöglicht uns hingegen den Gain durch die Zitationswertung des zusammengefügten Artikels und der entstehenden Teilartikel auszudrücken.

Lemma 1. *Gegeben eine Partition Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$ und ein zu extrahierender atomarer Artikel $a \in W$, dann gilt*

a) $\text{gain}(\mathcal{P}, a) = \mu(M_{a,\mathcal{P}} \setminus \{a\}) + \mu(\{a\}) - \mu(M_{a,\mathcal{P}})$,

b) $\text{gain}(\mathcal{P}, a) = |\,(\bigcup_{v \in M_{a,\mathcal{P}} \setminus \{a\}} N_D^{\text{in}}(v)\,) \cap N_D^{\text{in}}(a)\,|\,.$

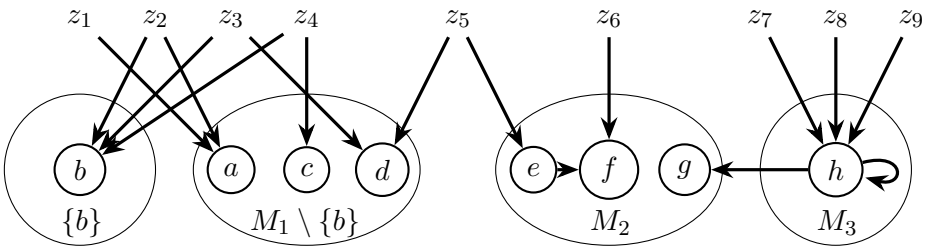
Beweis. Zuerst beweisen wir Lemma 1a. Den Ausdruck

$$\text{gain}(\mathcal{P}, a) \underset{\text{Def. Gain}}{=} \mu(\mathcal{P}_a) - \mu(\mathcal{P}) \underset{\text{Def. } \mu(\mathcal{P})}{=} \sum_{M \in \mathcal{P}_a} \mu(M) - \sum_{M \in \mathcal{P}} \mu(M)$$

erhält man durch simples Anwenden der Definitionen vom Gain und μ für Partitionen. Da die Mengen $\mathcal{P}_a$ und $\mathcal{P}$ sich lediglich darin unterscheiden, dass $M_{a,\mathcal{P}}$ nur in $\mathcal{P}$ enthalten ist und es in $\mathcal{P}_a$ durch $M_{a,\mathcal{P}} \setminus \{a\}$ und $\{a\}$ ersetzt wird, wirkt sich dies auf die Differenz ihrer Zitationswertungen so aus, dass gilt

$$\sum_{M \in \mathcal{P}_a} \mu(M) - \sum_{M \in \mathcal{P}} \mu(M) = \mu(M_{a,\mathcal{P}} \setminus \{a\}) + \mu(\{a\}) - \mu(M_{a,\mathcal{P}}).$$

$$\mathcal{P}_b = \{\{b\}, \{a, c, d\}, \{e, f, g\}, \{h\}\}$$



	$\{b\}$	$M_1 \setminus \{b\}$	M_2	M_3	$\mathcal{P}$
μ	3	5	4	4	16

Abbildung 3: Beispiel-Zitationsgraph aus Abbildung 2 nach der Extraktion des atomaren Artikels b.

Lemma 1b zeigen wir mithilfe von Lemma 1a. Wir wissen dadurch nämlich, dass

$$\text{gain}(\mathcal{P}, a) = \mu(M_{a,\mathcal{P}} \setminus \{a\}) + \mu(\{a\}) - \mu(M_{a,\mathcal{P}})$$

ist. Wenden wir nun jedoch die Definition von Unioncite auf die einzelnen Artikel an, erhalten wir

$$\mu(M_{a,\mathcal{P}} \setminus \{a\}) + \mu(\{a\}) - \mu(M_{a,\mathcal{P}}) \underset{\text{Def. } \mu(M)}{=} |\bigcup_{v \in M_{a,\mathcal{P}} \setminus \{a\}} N_D^{\text{in}}(v)| + |N_D^{\text{in}}(a)| - |\bigcup_{v \in M_{a,\mathcal{P}}} N_D^{\text{in}}(v)|.$$

Und da $M_{a,\mathcal{P}}$ von allen atomaren Artikeln zitiert wird die auch die beiden Teilartikel $M_{a,\mathcal{P}} \setminus \{a\}$ und $\{a\}$ zitieren, die Summe der Zitationswertungen der beiden Teilartikel aber die zitierenden Artikel doppelt wertet die beide zitieren, folgt

$$|\bigcup_{v \in M_{a,\mathcal{P}} \setminus \{a\}} N_D^{\text{in}}(v)| + |N_D^{\text{in}}(a)| - |\bigcup_{v \in M_{a,\mathcal{P}}} N_D^{\text{in}}(v)| = |(\bigcup_{v \in M_{a,\mathcal{P}} \setminus \{a\}} N_D^{\text{in}}(v)) \cap N_D^{\text{in}}(a)|.$$

□

Betrachten wir nun die Gains der atomaren Artikel aus unserem vorherigen Beispiel in Abbildung 2, ist zu erkennen, dass der g-Index erhöht werden kann. Durch Extraktion von Artikel b, der einen *Gain* von drei aufweist, gäbe es nämlich vier Artikel mit $\mu(\text{top}(\mathcal{P}_b), 4) = 16 \geq 4^2$, womit $\mathcal{P}_b$ einen g-Index von vier hätte (siehe Abbildung 3). Der h-Index bleibt aber unverändert, da $\mu(\{b\}) < 4$. Schaut man sich nun die einzelnen *Gains* der atomaren Artikel aus $M_{b,\mathcal{P}}$ *nach* der Extraktion (also in P_b) an, dann erkennt man eine Abnahme dieser (in diesem Fall für alle

auf null). Allgemein folgt aus Lemma 1b, dass der Gain eines atomaren Artikels v sich durch Extraktionen $F \in \mathcal{F}$ um jeweils eins verschlechtert, für jeden atomaren Artikel z_i, der neben v auch andere atomare Artikel in $M_{v,\mathcal{P}}$ zitiert hat, nach den Extraktionen (in $M_{v,\mathcal{P}_F}$) aber nur noch v zitiert. Diese Beeinflussung der Gains durch die Extraktionen, verkompliziert das Finden der maximierenden Extraktionsfolge bezüglich des g-Indexes. Hinzu kommt, dass im Gegensatz zu unserem Beispiel, die Top g Artikel bereits befüllt sein können. Das wiederum eröffnet die Frage, ob die entstehenden Teilartikel überhaupt in den Top g von $\mathcal{P}_v$ enthalten sind. Selbst wenn dies der Fall ist muss immer noch die Abwägung gemacht werden, ob eine Verdrängung von anderen Artikeln sich lohnt. Der *Gain* beschreibt also nicht den echten Zitationsanstieg, der in den g Top-Artikeln durch die Extraktion von v erreicht wird. Diesen beschreibt der *Realgain*.

3.2 Realgain

Definition (Realgain)**.** *Der Realgain eines atomaren Artikels v und einer Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$ beschreibt den Zitationanstieg, der durch die Extraktion von v aus $M_{v,\mathcal{P}}$ in den g Artikel, mit der größten Zitationswertung erzielt werden kann.*

$$\text{realgain}(\mathcal{P}, v, g) := \mu(\text{top}(\mathcal{P}_v, g)) - \mu(\text{top}(\mathcal{P}, g))$$

Um unseren Algorithmus für die maximierende Folge finden zu können, werden wir zuerst die Eigenschaften des *Realgain* einer einzelnen Extraktion genauer betrachten. Später werden wir die dabei erhaltenen Erkenntnisse auf ganze Extraktionsfolgen anwenden. Zuvor müssen wir aber noch die Verhältnisse der bei einer Extraktion entstehenden Teilartikel zu ihrem zusammengefügten Artikel und ihrem *Gain* vor der Extraktion durch ein Lemma genauer beschreiben. Lemma 2 besagt zum einen, dass die Teilartikel, die bei einer Extraktion entstehen, beide eine geringere oder gleiche Zitationswertung als den Artikel haben, aus dem extrahiert wird. Der *Gain* dieser Extraktion wiederum hat einen geringe oder gleiche Wertung als jeweils beide Teilartikel. Dieses Lemma wird in mehreren der folgenden Lemmas benötigt.

Lemma 2. *Gegeben eine Partition Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$ und ein zu extrahierender atomarer Artikel $a \in W$, dann gilt*

a) $\mu(M_{a,\mathcal{P}}) \geq \mu(M_{a,\mathcal{P}} \setminus \{a\})$ *und* $\mu(M_{a,\mathcal{P}}) \geq \mu(\{a\})$,

b) $\mu(M_{a,\mathcal{P}} \setminus \{a\}) \geq \text{gain}(\mathcal{P}, a)$ *und* $\mu(\{a\}) \geq \text{gain}(\mathcal{P}, a)$.

Beweis. Wir zeigen zuerst Lemma 2a. Durch anwenden der Definition von Unioncite erhält man die Gleichung

$$\mu(M_{a,\mathcal{P}}) \underset{\text{Def. } \mu(M)}{=} \Big| \bigcup_{v \in M_{a,\mathcal{P}}} N_D^{\text{in}}(v) \Big| .$$

Da $M_{a,\mathcal{P}} \setminus \{a\}$ und $\{a\}$ Teilmengen von $M_{a,\mathcal{P}}$ sind, muss dies auch für die Menge der atomaren Artikel gelten, die diese zitieren. Damit folgt für ihre Zitationswertungen, dass sie kleiner gleich

der, von $M_{a,\mathcal{P}}$ sind . In Gleichungen,

$$\mu(M_{a,\mathcal{P}}) \underset{\text{Def. } \mu(M)}{=} |\bigcup_{v \in M_{a,\mathcal{P}}} N_D^{\text{in}}(v)| \underset{\text{Teilmenge}}{\geq} |\bigcup_{v \in M_{a,\mathcal{P}} \setminus \{a\}} N_D^{\text{in}}(v)| \underset{\text{Def. } \mu(M)}{=} \mu(M_{a,\mathcal{P}} \setminus \{a\}), \text{ sowie}$$

$$\mu(M_{a,\mathcal{P}}) \underset{\text{Def. } \mu(M)}{=} |\bigcup_{v \in M_{a,\mathcal{P}}} N_D^{\text{in}}(v)| \underset{\text{Teilmenge}}{\geq} |N_D^{\text{in}}(a)| \underset{\text{Def. } \mu(M)}{=} \mu(\{a\}).$$

Als nächstes Beweisen wir die Aussage von Lemma 2b. Erneut wenden wir die Definition von Unioncite an. Danach nutzen wir aus, dass die Schnittmenge einer Menge mit einer beliebigen anderen Menge eine Teilmenge dieser Ursprungsmengen darstellt. Konkret bedeutet das hier, dass der Schnitt der zwei Mengen der zitierenden atomaren Artikel, die jeweils in den Teilartikeln $\{a\}$ und $M_{a,\mathcal{P}} \setminus \{a\}$ zitieren kleiner ist, als diese zwei Mengen vor dem Schnitt. Da der Gain nach Lemma 1b gleich dieser Schnittmenge ist haben wir gezeigt, dass

$$\mu(M_{a,\mathcal{P}} \setminus \{a\}) \underset{\text{Def. } \mu(M)}{=} |\bigcup_{v \in M_{a,\mathcal{P}} \setminus \{a\}} N_D^{\text{in}}(v)| \underset{\text{Teilmenge}}{\geq} |(\bigcup_{v \in M_{a,\mathcal{P}} \setminus \{a\}} N_D^{\text{in}}(v)) \cap N_D^{\text{in}}(a)| \underset{\text{Lemma 1b}}{=} \text{gain}(\mathcal{P}, a),$$

sowie

$$\mu(\{a\}) \underset{\text{Def. } \mu(M)}{=} |N_D^{\text{in}}(a)| \underset{\text{Teilmenge}}{\geq} |(\bigcup_{v \in M_{a,\mathcal{P}} \setminus \{a\}} N_D^{\text{in}}(v)) \cap N_D^{\text{in}}(a)| \underset{\text{Lemma 1b}}{=} \text{gain}(\mathcal{P}, a) \text{ gilt.}$$

□

Lemma 3 behandelt die verschieden Fälle, die den *Realgain* bestimmen und ermöglicht uns den *Realgain* eines atomaren Artikels a danach zu bestimmen, ob $M_{a,\mathcal{P}}$ *vor der Extraktion* in den Top g enthalten ist und ob die entstehenden Teilartikel es *nach der Extraktion* in die Top g geschafft haben.

Lemma 3. *Gegeben eine Partition Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$, ein zu extrahierender atomarer Artikel $a \in W$ und ein $g \in \mathbb{N}$, dann lässt sich die Berechnung des Realgain von a in $\mathcal{P}$ in die Fälle unterteilen*

a) *falls $M_{a,\mathcal{P}} \notin \text{top}(\mathcal{P}, g)$, dann folgt* $\text{realgain}(\mathcal{P}, a, g) = 0$,

b) *falls $M_{a,\mathcal{P}} \in \text{top}(\mathcal{P}, g) \wedge M_{a,\mathcal{P}} \setminus \{a\} \notin \text{top}(\mathcal{P}_a, g) \wedge \{a\} \notin \text{top}(\mathcal{P}_a, g)$, dann folgt* $\text{realgain}(\mathcal{P}, a, g) = \mu(\text{topGet}(\mathcal{P}, g+1, 1)) - \mu(M_{a,\mathcal{P}})$,

c) *falls $M_{a,\mathcal{P}} \in \text{top}(\mathcal{P}, g) \wedge M_{a,\mathcal{P}} \setminus \{a\} \in \text{top}(\mathcal{P}_a, g) \wedge \{a\} \notin \text{top}(\mathcal{P}_a, g)$, dann folgt* $\text{realgain}(\mathcal{P}, a, g) = \mu(M_{a,\mathcal{P}} \setminus \{a\}) - \mu(M_{a,\mathcal{P}})$,

d) *falls $M_{a,\mathcal{P}} \in \text{top}(\mathcal{P}, g) \wedge M_{a,\mathcal{P}} \setminus \{a\} \notin \text{top}(\mathcal{P}_a, g) \wedge \{a\} \in \text{top}(\mathcal{P}_a, g)$, dann folgt* $\text{realgain}(\mathcal{P}, a, g) = \mu(a) - \mu(M_{a,\mathcal{P}})$,

e) *falls $M_{a,\mathcal{P}} \in \text{top}(\mathcal{P}, g) \wedge M_{a,\mathcal{P}} \setminus \{a\} \in \text{top}(\mathcal{P}_a, g) \wedge \{a\} \in \text{top}(\mathcal{P}_a, g)$, dann folgt* $\text{realgain}(\mathcal{P}, a, g) = \text{gain}(\mathcal{P}, a) - \mu(\text{topGet}(\mathcal{P}, g, 1))$.

Beweis. Zuerst behandeln wir Lemma 3a. Wegen Lemma 2a gilt, dass $M_{a,\mathcal{P}} \setminus \{a\} \notin \text{top}(\mathcal{P}_a, g) \wedge \{a\} \notin \text{top}(\mathcal{P}_a, g)$ und damit $\text{top}(\mathcal{P}, g) = \text{top}(\mathcal{P}_a, g)$, denn wenn die Zitationswertung von $M_{a,\mathcal{P}}$ nicht für die Top g Artikel ausreicht, dann gilt dies auch für seine Teilartikel, die nach Lemma 2a kleinere Zitationswertungen aufweisen. Damit bleiben die Top g Artikel unverändert.

Als nächstes zeigen wir Lemma 3b. Es gilt $\text{top}(\mathcal{P}_a, g) = \text{top}(\mathcal{P}, g+1) \setminus \{M_{a,\mathcal{P}}\}$, da in diesem Fall $M_{a,\mathcal{P}} \in \text{top}(\mathcal{P}, g)$, aber beide Teilartikel nicht in den Top g sind. $M_{a,\mathcal{P}}$ wird also von dem nächst größten Artikel, der zuvor außerhalb von den Top g war, in diesen ersetzt.

Lemma 3c und Lemma 3d behandeln wir aufgrund der Ähnlichkeit der beiden Fälle gemeinsam. Es gilt hier $\text{top}(\mathcal{P}_a, g) = (\text{top}(\mathcal{P}, g) \setminus \{M_{a,\mathcal{P}}\}) \cup \{t\}$ für den Teilartikel $t \in \{M_{a,\mathcal{P}} \setminus \{a\}, \{a\}\}$, der es nach der Extraktion in die Top g geschafft hat, denn erneut geht $M_{a,\mathcal{P}}$ durch die Extraktion aus der Partition verloren, im Gegensatz zum vorherigen Fall, wird hier $M_{a,\mathcal{P}}$ aber von einem der Teilartikel t in den Top g ersetzt und nicht von dem nächst-größten Artikel außerhalb der Top g. Es sollte angemerkt werden, dass diese beiden Fälle wegen Lemma 2a stets einen Realgain kleiner gleich null aufweisen müssen.

Zuletzt beweisen wir Lemma 3e. Es gilt $\text{top}(\mathcal{P}_a, g) = (\text{top}(\mathcal{P}, g) \setminus \{M_{a,\mathcal{P}}, \text{topGet}(\mathcal{P}, g, 1)\}) \cup \{M_{a,\mathcal{P}} \setminus \{a\}, \{a\}\}$. Wieder geht $M_{a,\mathcal{P}}$ durch die Extraktion aus den Top g verloren, da jedoch beide entstehenden Teilartikel in den Top g enthalten sind, muss zusätzlich der kleinste Artikel von den g größten vor der Extraktion verdrängt werden. Alle restlich Artikel in den Top g bleiben nach der Extraktion darin erhalten. Es folgt

$$\text{realgain}(\mathcal{P}, a, g) = \mu(\text{top}(\mathcal{P}_a, g)) - \mu(\text{top}(\mathcal{P}, g)) = \mu(M_{a,\mathcal{P}} \setminus \{a\}) + \mu(\{a\}) - \mu(M_{a,\mathcal{P}}) - \mu(\text{topGet}(\mathcal{P}, g, 1)).$$

Dank Lemma 1a können wir dann durch den Gain ersetzen und es folgt

$$\mu(M_{a,\mathcal{P}} \setminus \{a\}) + \mu(\{a\}) - \mu(M_{a,\mathcal{P}}) - \mu(\text{topGet}(\mathcal{P}, g, 1)) \underset{\text{Lemma 1a}}{=} \text{gain}(\mathcal{P}, a) - \mu(\text{topGet}(\mathcal{P}, g, 1)).$$

□

Lemma 4 baut auf Lemma 3 auf. Lemma 4a beschreibt die Erkenntnis, dass der in Lemma 3e behandelte Fall eine notwendige Bedingung für einen *Realgain* größer als null darstellt. Lemma 4b besagt, dass der *Realgain* größer als null ist, *genau dann wenn* vor der Extraktion der zu extrahierende atomare Artikel einen *Gain* besitzt, der größer als der kleinste Artikel in den Top g ist.

Lemma 4. *Gegeben eine Partition Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$, ein zu extrahierender atomarer Artikel $a \in W$ und ein $g \in \mathbb{N}$, dann gilt:*

a) $\text{realgain}(\mathcal{P}, a, g) > 0 \Rightarrow M_{a,\mathcal{P}} \setminus \{a\} \in \text{top}(\mathcal{P}_a, g) \wedge \{a\} \in \text{top}(\mathcal{P}_a, g)$

b) $\text{gain}(\mathcal{P}, a) > \mu(\textit{topGet}(\mathcal{P}, g, 1)) \Leftrightarrow \text{realgain}(\mathcal{P}, a, g) > 0$

Beweis. Lemma 4a lässt sich durch die Fallunterscheidung von Lemma 3 zeigen. Außer dem Fall, dass $M_{a,\mathcal{P}} \setminus \{a\} \in \text{top}(\mathcal{P}_a, g)$ und $a \in \text{top}(\mathcal{P}_a, g)$ ist (Lemma 3e), ist der *Realgain* nämlich sonst

immer kleiner gleich null, da $M_{a,\mathcal{P}}$ nach der Extraktion entweder mit einem Artikel ersetzt wurde, der keine größere Zitationswertung haben kann (Lemma 3b bis 3d), oder es zu keiner Veränderung in den Top g kommt (Lemma 3a). In den Fällen in denen $M_{a,\mathcal{P}}$ ersetzt wird nehmen entweder der größte Artikel außerhalb der Top g vor der Extraktion, oder ein durch die Extraktion entstehender Teilartikel von $M_{a,\mathcal{P}}$ seinen Platz ein. Die Teilartikel sind wegen Lemma 2a kleiner oder gleich $M_{a,\mathcal{P}}$ und die Artikel von außerhalb sind es, da sie vor der Extraktion sonst wie $M_{a,\mathcal{P}}$ in den Top g gewesen sein mussten.

Als nächstes beweisen wir Lemma 4b. Wir beginnen hierbei mit der Hinrichtung. Wegen Lemma 2b wissen wir bereits, dass die Zitationswertungen der entstehenden Teilartikel $M_{a,\mathcal{P}} \setminus \{a\}$ und $\{a\}$ der Extraktion von a mindestens so groß sind, wie sein Gain. Also ist

$$\mu(M_{a,\mathcal{P}} \setminus \{a\}),\ \mu(\{a\}) \underset{\text{Lemma 2b}}{\geq} \text{gain}(\mathcal{P}, a).$$

Aus der Prämisse der Hinrichtung $\text{gain}(\mathcal{P}, a) > \mu(\mathit{topGet}(\mathcal{P}, g, 1))$ folgt dadurch

$$\mu(M_{a,\mathcal{P}} \setminus \{a\}),\ \mu(\{a\}) \underset{\text{Lemma 2b}}{\geq} \text{gain}(\mathcal{P}, a) > \mu(\text{topGet}(\mathcal{P}, g, 1)).$$

Der zusammengefügte Artikel $M_{a,\mathcal{P}}$ aus dem Extrahiert wurde, kann außerdem nicht selbst $\text{topGet}(\mathcal{P}, g, 1)$ sein, da sein Gain nicht größer als dessen Zitationswertung sein kann. Somit ersetzen die beiden Teilartikel beide diese zusammengefügten Artikel in $\text{top}(\mathcal{P}_a, g)$ und es gilt

$$M_{a,\mathcal{P}} \setminus \{a\} \in \text{top}(\mathcal{P}_a, g) \wedge \{a\} \in \text{top}(\mathcal{P}_a, g). \tag{1}$$

Dies entspricht dem in Lemma 3e behandelten Fall, weshalb

$$\text{realgain}(\mathcal{P}, a, g) = \text{gain}(\mathcal{P}, a) - \mu(\text{topGet}(\mathcal{P}, g, 1)) \tag{2}$$

folgt. Dabei wird $\mu(\text{topGet}(\mathcal{P}, g, 1))$ aus den Top g verdrängt. Durch die Prämisse $\text{gain}(\mathcal{P}, a) > \mu(\mathit{topGet}(\mathcal{P}, g, 1))$ der zu beweisenden Implikation folgt damit, dass die Konklusion wahr ist.

Als nächstes zeigen wir die Rückrichtung des Lemmas 4b. Aus Lemma 4a können wir durch die Prämisse der Rückrichtung $\text{realgain}(\mathcal{P}, a, g) > 0$ erneut schließen, dass (1) der Fall ist, und somit durch Lemma 3e auch (2). Da $\text{realgain}(\mathcal{P}, a, g) > 0$ muss damit die Konklusion $\text{gain}(\mathcal{P}, a) > \mu(\mathit{topGet}(\mathcal{P}, g, 1))$ gelten. □

Die Betrachtung einer einzelnen Extraktion reicht jedoch nicht aus, um unser Extraktions-Problem zu lösen, da die Gains durch die Extraktionen der anderen atomaren Artikel im selben atomaren Artikel beeinflusst werden. Deshalb werden wir im Folgendem mithilfe der bereits etablierten Definitionen neue Hilfsdefinitionen für ganze Extraktionsfolgen festlegen. Der *GainSeq* beschreibt dabei den *Gain* einer Folge von Extraktionen und damit die Summe der *Gains* der einzelnen Extraktionen innerhalb ihrer Extraktionsfolge. *RealGainSeq* hingegen, beschreibt die Summe der *Realgains* der einzelnen Extraktionen innerhalb ihrer Extraktionsfolge.

$$\text{gainSeq}(\mathcal{P}, (f_1, \ldots, f_k)) := \mu(\mathcal{P}_{(f_1,\ldots,f_k)}) - \mu(\mathcal{P}) = \sum_{i=1}^{k} \text{gain}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, f_i)$$

$$\text{realgainSeq}(\mathcal{P}, (f_1, \ldots, f_k), g) = \mu(\text{top}(\mathcal{P}_{(f_1,\ldots,f_k)}, g)) - \mu(\text{top}(\mathcal{P}, g)) = \sum_{i=1}^{k} \text{realgain}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, f_i, g)$$

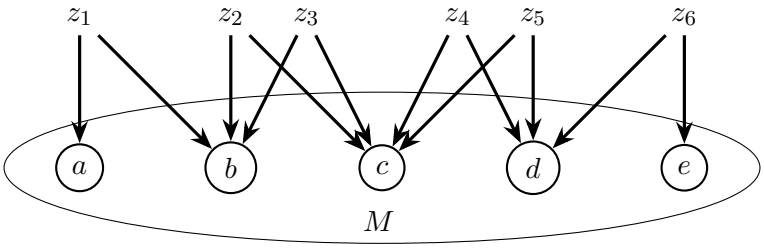
Der *GainInSeq* hingegen beschreibt den Gain der i-ten Extraktion innerhalb der Folge F aus $\mathcal{P}$, also nach der Extraktion aller vorherigen Artikel. *RealgainInSeq* beschreibt entsprechend den *Realgain*. Es handelt sich auch hier um Hilfsdefinitionen. Diese sollen die Lesbarkeit vereinfachen.

$$\text{gainInSeq}(\mathcal{P}, i, F) := \text{gain}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, f_i)$$

$$\text{realgainInSeq}(\mathcal{P}, i, F, g) := \text{realgain}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, f_i, g)$$

4 g-Index-Manipulation durch Artikel-Extraktion ist NP-schwer

Bevor wir durch einen Reduktionsbeweis zeigen werden, dass *g-Index-Manipulation durch Artikel-Extraktion* NP-schwer ist, werden wir kurz versuchen zu beschreiben, was das Problem so kompliziert macht. Wie wir bereits wissen beeinflussen die Extraktionen von atomaren Artikeln die Gains anderer atomaren Artikel im selben zusammengefügten Artikel. Die grundlegende Problematik hierbei erschließt sich beim betrachten von Folgen, die ausschließlich aus einem bestimmten zusammengefügten Artikel extrahieren. Solche Folgen lassen sich als Teilfolgen einer Gesamtfolge betrachten, die sich gegenseitig nicht beeinflussen, da die Gains aus anderen zusammengefügten Artikeln durch Extraktion unverändert bleiben. Sei $F_{M,j}$ eine solche Teilfolge, die ausschließlich aus dem zusammengefügten Artikel M extrahiert und den maximal erreichbaren GainSeq für genau j viele Extraktionen aus M besitzt (wir werden die Bezeichnung $F_{M,j}$ im Rest der Arbeit für eine solche Folge verwenden). Dann wäre es möglich, dass keine der möglichen Teilfolgen von M, die genau diese Eigenschaft für $j+1$ erfüllen auch alle Extraktionen von $F_{M,j}$ beinhalten. Beide diese Folgen können also komplett verschieden sein. Um dies beispielhaft zu verdeutlichen betrachten wir Abbildung 4. Hier sieht man einen zusammengefügten Artikel M für den die zu extrahierenden Artikel für einen maximalen GainSeq für genau eine Extraktion $F_{M,1}$ nicht in der für genau zwei Extraktionen $F_{M,2}$ enthalten sind. Für genau eine Extraktion würde man c mit einem Gain von vier Extrahieren, wodurch jedoch die Gains der anderen atomaren Artikel auf eins sinken. Alle Extraktionsfolgen der Länge zwei die c extrahieren hätten damit einen GainSeq von fünf. Extrahiert man jedoch b und d, die sich in den GainInSeqs nicht beeinflussen, da sie keine sie zitierenden Artikel miteinander teilen, erhält man einen GainSeq von sechs. Die Optimierung der ausgewählten Extraktionen eines zusammengefügten Artikels nach dem GainSeq ist damit von der Anzahl dieser Extraktionen abhängig und damit wahrscheinlich die hauptsächliche Ursache für die NP-Schwierigkeit des Problems. Aus der Betrachtungsebene aller Artikel stellt sich also die Frage, wie viele Extraktionen man aus welchem zusammengefügten Artikel durchführt. Dies ist zum einen davon abhängig, welche Zitationswertungen die Top g Artikel der betrachteten Partition $\mathcal{P}$ aufweisen, zum anderen davon was für maximale GainSeqs die möglichen Teilfolgen $F_{M,j}$, verschiedener Länge j, der zusammengefügten Artikel M erzielen können. Da das Problem NP-schwer ist, wie wir im Folgendenm zeigen werden, ist damit zu rechnen, dass das ermitteln der richtigen $F_{M,j}$, die zur RealgainSeq maximierenden Folge zusammengesetzt werden können, nicht



	a	b	c	d	e
Gain	1	3	4	3	1

Abbildung 4: Beispiel für einen zusammengefügten Artikel M für den die zu extrahierenden Artikel für einen maximalen GainSeq für genau eine Extraktion $F_{M,1}$ nicht in der für genau zwei Extraktionen $F_{M,2}$ enthalten sind.

in polynomieller Zeit lösbar. Wir werden uns deshalb damit abfinden zuerst die GainSeqs *aller* $F_{M,j}$ zu berechnen. Wie man aus diesen errechneten GainSeqs die RealgainSeqs berechnen kann, betrachten wir erst im nächsten Abschnitt.

Theorem 1. *g-Index-Manipulation durch Artikel-Extraktion ist NP-schwer*

Beweis. Bei dem Beweis von Theorem 1 führen wir *Independent Set auf drei-regulären Graphen* auf das *Entscheidungsproblem der g-Index-Manipulation durch Extraktion* zurück. *Independent Set auf drei-regulären Graphen* ist wie folgt defininiert:

Independent Set auf drei-regulären Graphen
Eingabe: Ein ungerichteter Graph $G' = (V', E')$, für den gilt, dass jeder Knoten aus G' genau drei Nachbarn hat, sowie eine natürliche Zahl k'.
Frage: Existiert ein $I \subseteq V'$ der Größe k', wobei in G' keine Kanten zwischen Knoten aus I existieren?

Die in der Problemdefinition eingeführte Teilmenge I eines Graphen bezeichnet man als *Independent Set* der Größe k. Da *Independent Set auf drei-regulären Graphen* NP-schwer ist, muss dies auch für ein Problem gelten, auf das wir es in polynomieller Zeit reduzieren, also unser g-Index-Problem [7]. Wir reduzieren hierbei präziser auf das Entscheidungs-Problem, *ob ein Folge existiert*, die den gewünschten g-Index hervorrufen kann. Die Folge an sich wird nicht gesucht. Wir können jedoch davon ausgehen, dass die Suche nach der eigentlichen Folge ebenfalls NP-schwer sein muss, da diese zur Lösung des Entscheidungsproblem verwendet werden kann. Die durch die Reduktion zu erzeugenden Instanz für das *Entscheidungsproblem der g-Index-Manipulation durch Extraktion* sind zum einen ein Zitationsgraph $D = (V, Z)$, eine Menge $W \subseteq V$ von Artikeln des Autors, eine Partition $\mathcal{P}$ von W und eine natürliche Zahl g, die den zu erreichenden g-Index angibt. Hierfür zur Verfügung steht eine Instanz G', k' von *Independent Set auf drei-regulären*

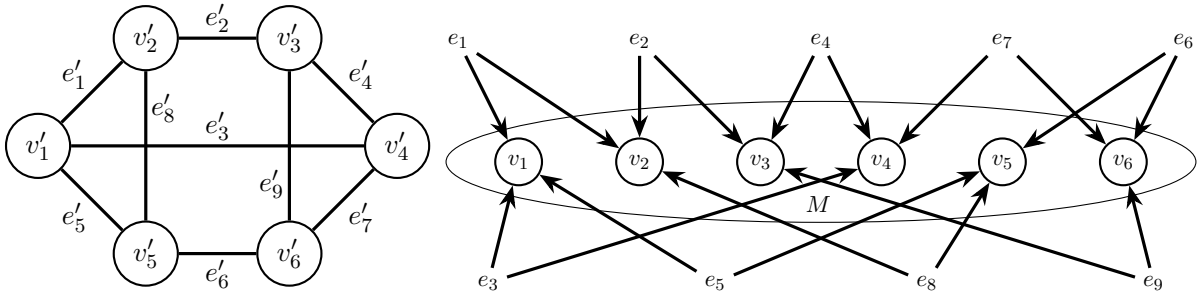


Abbildung 5: Ein Beispiel-drei-regulärer-Graph G' der sechs Knoten besitzt (links) und der entsprechende durch die Reduktion erzeugte zusammengefügte Artikel M. (rechts)

Graphen. Die Polynomialzeit-Reduktion ist damit folgende:

- Setze g als kleinste natürliche Zahl, sodass $g \geq k' + 2$ und $g^2 \geq |E'| + (g-1) \cdot 3$. Dieses g lässt durch inkrementelles durchprobieren finden, indem man beginnend von $g = k'+2$ hoch geht, bis die zweite Ungleichung erfüllt wird. Da die linke Seite dieser bezüglich g schneller wächst, findet man auf diese weise auch irgendwann ein solches g, welches auch polynomiell bezüglich der Eingabe groß ist.
- Erzeuge einen atomaren Artikel v_i für $i \in \{1, \ldots, |V'|\}$ mit $v_i \in V$ und $v_i \in W$ für jeden Knoten $v'_i \in V'$.
- Erzeuge auch einen Artikel $e_i \in V$ mit $i \in \{1, \ldots, |E'|\}$ für jede Kante $e'_i \in E'$. Dabei zitieren alle Artikel e aus Kanten in E' die zwei Artikel aus V, die die Knoten darstellen, die sie in V' verbinden.
- Erzeuge atomaren Artikel s_i für alle $i \in \{1, \ldots, g - k' - 1\}$ mit $s_i \in V$, sowie $s_i \in W$. Erstelle dann $g^2 - |E'| - 3 \cdot k'$ weitere atomare Artikel, die in V aber nicht in W enthalten sein sollen, welche jeweils ein s_i gleichmäßig verteilt zitieren. Die atomaren Artikel s_i bilden dann jeweils eigenen Artikel $\{s_i\} \in \mathcal{P}$. Wie wir weiter unten zeigen werden gilt für diese per Konstruktion $\mu(\{s_i\}) \geq 3$
- Füge in der Partition $\mathcal{P}$ alle atomaren Artikel v in einem Artikel $M \in \mathcal{P}$ zusammen. Für M gilt damit $\mu(M) = |E'|$. Jeder atomare Artikel in M hat dabei wegen der Drei-Regularität von G' drei Zitate und einen Gain von drei.

Die durch die Reduktion entstandene g-Index-Instanz, hat per Konstruktion k' „freie Plätze“ in den Top g Artikeln, da ein Platz von M und $g-k'-1$ Plätze bereits von den $\{s_i\}$ besetzt wurden. Das bedeutet, dass der GainInSeq der ersten k' Extraktionen ihrem RealgainInSeq entspricht

(siehe Lemma 3e). Außerdem benötigt die Summe der Zitationswertungen der Top g Artikel eine Erhöhung von genau $3 \cdot k'$, damit $\mu(\text{top}(\mathcal{P}_F, g)) \geq g^2$ erfüllt ist, da per Konstruktion gilt, dass

$$\mu(\mathcal{P}) = \mu(\{s_1\}) + \ldots + \mu(\{s_{g-k'-1}\}) + \mu(M) = g^2 - |E'| - 3 \cdot k' + |E'| = g^2 - 3 \cdot k'$$

ist. Wegen der Drei-Regularität von G' ist in der Startpartition $\mathcal{P}$ der Gain aller atomaren Artikel $v_i \in M$ gleich drei, weshalb weniger als k' Extraktionen für eine solchen Zitationswert-Erhöhung nicht ausreichen. Auch haben alle atomaren Artikel v_i in der Startpartition $\mathcal{P}$ eine Zitationswertung von drei. Die Artikel s_i haben alle nur einen atomaren Artikel mit einem Gain von null und sind deshalb für zitations-verbessernde Extraktionen unbrauchbar. Somit sind die v_i die einzigen atomaren Artikel in W, die in $\mathcal{P}$ einen Gain größer als null haben. Die s_i haben, wegen der gleichmäßigen Verteilung der $g^2 - |E'| - 3 \cdot k'$ Artikel die ausschließlich sie zitieren, jeweils per Konstruktion eine Zitationswertung von mindestens drei, da

$$g^2 - |E'| - 3 \cdot k' \underset{g^2 \geq |E'| + (g-1)\cdot 3}{\geq} |E'| + (g-1) \cdot 3 - |E'| - 3 \cdot k' = (g - k' - 1) \cdot 3$$

gilt. Aufgeteilt auf die $(g-k'-1)$ Artikel s_i, folgen die jeweiligen Zitationswertung von mindestens drei. Die durch die Extraktionen der Artikel $v_i \in M$ erzeugten Teilartikel $\{v_i\}$ besitzen wie bereits erwähnt ebenfalls eine Zitationswertung von drei. Dementsprechend hat auch M eine, da die v_i in ihr enthalten sind. Somit haben alle Artikel in $\mathcal{P}$ eine Zitationswertung von mindesten drei. Nach k' Extraktionen verschiedener v_i würde dies durch die Entstehung der k' Teilartikel $\{v_i\}$ bedeuten, dass dies damit auch für alle Artikel in den dadurch resultierenden Top g gilt. Nach Lemma 4b hätten damit alle darauffolgenden Extraktionen einen RealgainInSeq von kleiner oder gleich null, da die GainInSeqs der v_i nicht groß genug sind um die Zitationswertungen dieser Artikel zu übersteigen und sie aus den Top g zu verdrängen.

Da die v_i die einzigen atomaren Artikel mit einem positivem Gain in der Startpartition $\mathcal{P}$ sind, muss dies bedeuten, dass die einzige Möglichkeit g^2 zu erreichen ist, mindestens k' Artikel zu extrahieren, die einen GainInSeq von jeweils drei aufweisen. Da die GainInSeqs von atomaren Artikeln im Extraktionsverlauf jedoch sinken können, wenn alle atomaren Artikel extrahiert wurden mit dem sie sich einen zitierenden Artikel in ihrem zusammengefügten Artikel geteilt haben, ist es notwendig dass diese extrahierten v_i sich keine zitierenden atomaren Artikel teilen.

Wir beweisen nun mit der Hinrichtung der Reduktion. Angenommen es existiert ein Independent Set I in G' der Größe k'. Da es sich um ein Independent Set handelt, gibt es zwischen den v'_i keine Kanten $e'_i \in E'$. Damit gibt es keine zitierenden Artikel e_i in der g-Index-Instanz, die sich die entsprechenden $v_i \in M$ teilen. Somit würden die Artikel v_i, die diese Knoten $v'_i \in I$ in der g-Index-Instanz darstellen, sich durch Extraktion nicht gegenseitig in den GainInSeqs beinflussen. Da es zusätzlich ein drei-regulärer Graph ist, gilt damit, dass eine beliebige Folge die genau diese v_i extrahiert k' Extraktionen mit einem GainInSeq von jeweils drei hätte. Das g-Index-Problem wäre damit durch diese Folge erfüllt.

Nehmen wir nun für die Rückrichtung an, es gäbe eine Folge F die das g-Index-Problem erfüllt. Demnach muss F auch mindestens k' Artikel $v_i \in M$ extrahieren, die einen GainInSeq

von drei aufweisen und sich demnach keine zitierenden atomaren Artikel e_i teilen. Da zitierende Artikel e_i für Kanten e'_i in G' stehen, muss zwischen den Knoten $v'_i \in G'$, die diese extrahierten Artikel darstellen, ein Independent Set bestehen. □

5 Algorithmus

In diesem Abschnitt werden wir einen Algorithmus für das *g-Index-Manipulations*-Problem entwickeln. Zum Lösen des *g-Index-Manipulations*-Problem wird es unsere Strategie sein eine Folge zu ermitteln, die die Summe der Zitationswertungen in den g Artikeln mit den größten Zitationswertungen maximiert. Um Folgen mit Extraktionen die keinen Einfluss auf diese g haben auszuschließen, soll es zusätzlich eine kleinste dieser Folgen sein. Mithilfe des *RealgainSeq* lässt sich die gesuchte maximierende Folge nun als eine Folge F_{opt} von Extraktionen beschreiben, sodass $\text{realgainSeq}(\mathcal{P}, F_{opt}, g)$ maximal ist, ohne dass es eine kürzere Folge E mit $\text{realgainSeq}(\mathcal{P}, F_{opt}, g) = \text{realgainSeq}(\mathcal{P}, E, g)$ gibt.

Der Algorithmus lässt sich in drei Teile unterteilen. Der Erste ist das Preprocessing der *Teilfolgen* $F_{M,j}$, mit den maximal erreichbaren GainSeqs, die mit genau j Extraktion aus ausschließlich zusammengefügtem Artikel M zu erreichen sind. Ebenso werden in diesem Schritt die GainSeqs dieser $F_{M,j}$ selbst ermittelt. Im zweiten Teil setzen wir aus diesen Teilfolgen $F_{M,j}$ die *Gesamtfolgen* F^k der Länge k zusammen, die erneut für ihre Länge den maximal erreichbaren GainSeq aufweisen. Die F^k sind im Gegensatz zu den $F_{M,j}$ jedoch nicht auf Extraktionen einzelner zusammengefügter Artikel beschränkt und sind damit für *alle* möglichen Extraktionsfolgen der Länge k aus $\mathcal{F}$ die Folgen mit den maximalen GainSeqs. Im dritten und letzten Teil berechnen wir die RealgainSeqs dieser F^k und finden heraus, bei welchen von ihnen es sich um die gesuchten Folge F_{opt} mit dem maximalen RealgainSeq handelt. Wir beginnen zuerst mit der Beschreibung von Algorithmus 1, dem Preprocessing.

5.1 Preprocessing der Teilfolgen $F_{M,j}$

Wir berechnen im ersten Teilalgorithmus (Algorithmus 1) für alle Artikel $M_i \in \mathcal{P}$ mit $i \in \{1, \ldots, |\mathcal{P}|\}$ und für alle Teilfolgen-Längen $j \in \{1, \ldots, \min(g, |M_i|)\}$ die Teilfolgen $F_{M_i,j}$. Die Teilfolge $F_{M_i,j}$ ist eine Folge der Länge j, die ausschließlich atomare Artikel aus M_i extrahiert und für die gilt, dass es keine andere Folge S aus j Extraktionen aus M_i gibt mit $\text{gainSeq}(\mathcal{P}, S) > \text{gainSeq}(\mathcal{P}, F_{M_i,j})$. Diese ermitteln wir, indem im Prinzip einfach alle GainSeqs der möglichen Teilfolgen errechnet und verglichen werden.

Dieser Teilalgorithmus nutzt aus, dass wir die Reihenfolge einer Folge von Extraktionen vertauschen können, ohne dass sich die entstehende Partition, ihr GainSeq oder ihr RealgainSeq verändert. Somit müssen wir nicht die verschiedenen Reihenfolgen der zu extrahierenden Artikel unterscheiden. Wir beschreiben dies in Lemma 5.

Lemma 5. *Gegeben eine Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$, ein $g \in \mathbb{N}$, sowie zwei Folgen $F, E \in \mathcal{F}$ mit $F := (f_1, \ldots, f_k)$ und $E := (e_1, \ldots, e_k)$, wobei*

Algorithm 1: Preprocessing der Teilfolgen $F_{M,j}$

```
1  for M ∈ 𝒫 do
2      for (z, v) ∈ E do
3          if v ∈ M then                    // citIn und citBy werden als ∅ initialisiert
4              citIn(z, M) ← citIn(z, M) ∪ v          // citIn(z, M) = {v | v ∈ M ∧ (z, v) ∈ E}
5              citBy(M) ← citBy(M) ∪ z                // citBy(M) = {z | ∃v ∈ M.(z, v) ∈ E}
6      for v ∈ M do
7          gain(v) ← 0
8      for z ∈ citBy(M) do
9          if |citIn(z, M)| > 1 then
10             for v ∈ citIn(z, M) do
11                 gain(v) ← gain(v) + 1
12 for M ∈ 𝒫 do
13     for T ⊆ M, mit |T| ≤ min(g, |M|) do
14         for v ∈ T do
15             gainIS(v) ← gain(v)
16         for z ∈ citBy(M) do
17             citInIS(z) ← citIn(z, M)
18         Rest ← T
19         T_gainSeq ← 0
20         T_Seq wird geleert
21         for i ∈ {1, ..., |T|} do
22             e ← v ∈ Rest mit größtem gainIS(v)
23             Rest ← Rest \ {e}
24             T_Seq[i] ← (e, gainIS(e))
25             T_gainSeq ← T_gainSeq + gainIS(e)
26             for z ∈ citBy(M) do
27                 if e ∈ citInIS(z) ∧ |citInIS(z)| = 2 then
28                     citInIS(z) ← citInIS(z) \ {e}
29                     for v ∈ citInIS(z) do
30                         gainIS(v) ← gainIS(v) − 1
31                 else
32                     citInIS(z) ← citInIS(z) \ {e}
33         if T_gainSeq ≥ maxGainSeq_{M,|T|} then
34             maxGainSeq_{M,|T|} ← T_gainSeq     // alle maxGainSeq initialisieren mit 0
35             F_{M,|T|} ← T_Seq
```

$\{f_1, \ldots, f_k\} = \{e_1, \ldots, e_k\}$, *dann gilt*

a) $\mathcal{P}_F = \mathcal{P}_E$,

b) $\text{gainSeq}(\mathcal{P}, F) = \text{gainSeq}(\mathcal{P}, E)$ *und*

c) $\text{realgainSeq}(\mathcal{P}, F, g) = \text{realgainSeq}(\mathcal{P}, E, g)$.

Beweis. Lemma 5a lässt sich direkt aus der Definition von Extraktionen entnehmen. Es gilt

$$\mathcal{P}_{(f_1,\ldots,f_k)} = \bigcup_{M \in \mathcal{P}} (\{M \setminus \{f_1, \ldots, f_k\}\}) \cup \{\{f_1\}, \ldots, \{f_k\}\} = \mathcal{P}_{(e_1,\ldots,e_k)},$$

da $\{f_1, \ldots, f_k\} = \{e_1, \ldots, e_k\}$ ist. Lemma 5b und Lemma 5c folgen damit aus Lemma 5a □

Die Zeilen in die wir uns in diesem Teilabschnitt beziehen stammen aus Algorithmus 1. In den Zeilen 2 bis 5 merken wir uns in den $\text{citIn}(z, M)$ welche atomaren Artikel z welche anderen atomaren Artikel in einem bestimmten zusammengefügten Artikel M zitieren. Dabei merkt sich $\text{citBy}(M)$ welche dieser atomaren Artikel z mindestens einmal M zitieren, was uns ermöglicht später lediglich die relevanten Artikel, die auch M zitieren zu betrachten. Von den Zeilen 6 bis 11 berechnen wir alle Gains der atomaren Artikel v der zusammengefügten Artikel M, der nach Lemma 1b der Anzahl der Artikel entspricht, die sowohl v, als auch mindestens einen anderen atomaren Artikel in M zitieren.

Von den Zeilen 12 bis 35 gehen wir für alle zusammengefügte Artikel M über alle Teilmengen T dieser, die kleiner oder gleich g sind und ermitteln jeweils eine Extraktionsfolge T_{Seq} aus diesen T, mit kleiner werdenden oder gleichbleibenden GainInSeqs. Dass die Teilfolgen monoton fallend bezüglich der GainInSeqs angeordnet sind, wird im dritten Teilalgorithmus von Bedeutung sein. Die Extraktionsfolge T_{Seq} ist ein Array aus Tupeln, mit dem i-ten extrahierten Artikel im ersten und dem GainInSeq der i-ten Extraktion im zweiten Eintrag. Für alle zusammengefügte Artikel M merken wir uns allerdings nur die T_{Seq}, mit dem größten GainSeq T_{gainSeq} für alle Folgen aus Teilartikeln von M, der selben Länge j. Diese speichern wir in den $F_{M,j}$. Um die T_{gainSeq} und T_{Seq} zu ermitteln, simulieren wir die Extraktionsfolge von Zeile 21 bis 32 mithilfe der zuvor ermittelten *Gains*, *citIn* und *citBy*. Abhängig von den bereits vollzogenen Extraktionen, speichern wir ihren momentanen Zustand hierfür in den temporären Variablen *gainIS* und *citInIS* (IS steht für das englische „in sequence“). Bei jedem Extraktionsschritt entfernen wir dabei den extrahierten Artikel aus den *citInIS*. Immer wenn zitierende Artikel durch die aktuelle Extraktion nur noch einen einzigen atomaren Artikel v in M zitieren, reduzieren wir den Gain von diesem v um eins (Zeile 27 bis 30). Von den Zeile 14 bis 18 setzen wir diese temporären Variablen, zusammen mit der Menge der noch nicht extrahierten Artikeln *Rest*, T_{Seq} und T_{gainSeq} auf den Zustand vor der ersten Extraktion zurück, um den Ablauf für eine neue Teilmenge T des aktuellen M zu wiederholen.

5.2 Zusammensetzen der F^k mit maximalen GainSeqs für k Extraktionen

Als nächstes folgt der zweite Teilalgorithmus (Algorithmus 2), der durch die zuvor errechneten $F_{M,j}$ die Teilfolgen F^k mit den maximalen GainSeqs aller Folgen mit k Extraktionen ermittelt. Er basiert auf folgender rekursiven Formel, wobei die Werte nach dem Prinzip der dynamischen Programmierung durch die Ergebnisse vorheriger Rechnungen ermittelt werden:

$$R[m][j][k] = \begin{cases} 0 & \text{, falls } m \leq 0 \vee k \leq 0 \\ R[m][j-1][k] & \text{, falls } k < j \vee |M| < j \\ \max(\text{maxGainSeq}_{M,j} + R[m-1][g][k-j], R[m-1][g][k]) & \text{, falls } j = 1 \\ \max(\text{maxGainSeq}_{M,j} + R[m-1][g][k-j], R[m][j-1][k]) & \text{, sonst} \end{cases}$$

In diesem Algorithmus wurden alle zusammengefügten Artikel M mit einer Zahl $m \in \{1, \ldots, |\mathcal{P}|\}$ nummeriert, die sie kennzeichnen. *$R[m][j][k]$ beschreibt dabei den maximalen GainSeq, den man mit k Extraktionen erreichen kann, indem man nur die zusammengefügten Artikel mit Kennzahlen von eins bis m betrachtet und höchstens j atomare Artikel aus dem aktuellen m-ten zusammengefügten Artikel extrahiert.* In $S[m][j][k]$ wird gespeichert welche Teilfolgen $F_{M,j}$ verwendet wurden, um den GainSeq aus $R[m][j][k]$ zu erhalten. Gespeichert wird als Tupel der Indizes M und j. R ist als dreidimensionale $|P| \times g \times g$ Matrix umsetzbar. Seine Dimensionen sollten in der Reihenfolge der Nummerierung der zusammengefügten Artikel M, der maximalen Anzahl j der Extraktionen aus dem aktuellen M und schließlich der Anzahl der Extraktionen insgesamt iteriert werden.

Dieser Teilalgorithmus nutzt aus, dass Extraktionen von atomaren Artikeln aus verschieden zusammengefügte Artikeln sich in den GainInSeqs nicht gegenseitig beeinflussen, was direkt aus Lemma 1b zu entnehmen ist. Der Gain eines atomaren Artikels v entspricht nämlich der Anzahl der zitierenden Artikel, die sowohl v als auch mindestens einen anderen Artikel in $M_{v,\mathcal{P}}$ zitieren. Extraktionen aus einem zusammengefügten Artikel verändern jedoch nicht die anderen zusammengefügte Artikel in der Partition, oder deren Zitate. Da die $F_{M,j}$ bereits die maximalen GainSeqs $\text{maxGainSeq}_{M,j}$ aller Teilfolgen besitzen, die ausschließlich j Extraktionen aus den jeweiligen M vollziehen, müssen wir zum Finden der F^k nur noch für alle k die Kombinationen dieser finden, sodass die Summen ihrer GainSeqs maximal sind. Dabei müssen die Längen aller Teilfolgen $F_{M,j}$ der Kombinationen zusammen höchstens k ergeben. Diese Kombinationen an Teilfolgen mit maximalen Summen an GainSeqs bilden dann die Gesamtfolgen F^k. Wir finden sie dann letztendlich in den Einträgen $S[|\mathcal{P}|][g][k]$ für $k \in \{1, \ldots, g\}$, da hier bereits alle zusammengefügten Artikel M betrachtet wurden und die maximale Anzahl $j = g$ an Artikeln aus diesen extrahiert werden dürfen. Deren GainSeqs sind in den entsprechenden Einträgen von R zu finden.

Um die Korrektheit des Teilalgorithmus zu zeigen, müssen wir beweisen, dass die rekursive Gleichung auf die er basiert auch das beschreibt was sie soll.

Lemma 6. *Gegeben eine Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$ und $\text{maxGainSeq}_{M,j}$ für $M \in \mathcal{P}$ der maximale GainSeq den man mit j Extraktionen aus M erreichen kann, dann beschreibt $R[m][j][k]$ den maximalen GainSeq, den man mit k Extraktionen*

Algorithm 2: Zusammensetzen der Gesamtfolgen F^k mit maximalem GainSeq für k Extraktionen

```
1  for m ∈ {0, ..., |P|} do
2      for j ∈ {1, ..., g} do
3          for k ∈ {1, ..., g} do
4              if m ≤ 0 ∨ k ≤ 0 then
5                  R[m][j][k] = 0
6                  S[m][j][k] = ∅
7              else if k < j ∨ |M| < j then    // Das M, dass mit m nummeriert wurde
8                  R[m][j][k] = R[m][j − 1][k]
9                  S[m][j][k] = S[m][j − 1][k]
10             else if j = 1 then
11                 if maxGainSeq_{M,j} + R[m − 1][g][k − j] > R[m − 1][g][k] then
12                     R[m][j][k] = maxGainSeq_{M,j} + R[m − 1][g][k − j]
13                     S[m][j][k] = {(M, j)} ∪ S[m − 1][g][k − j]
14                 else
15                     R[m][j][k] = R[m − 1][g][k]
16                     S[m][j][k] = S[m − 1][g][k]
17             else
18                 if maxGainSeq_{M,j} + R[m − 1][g][k − j] > R[m][j − 1][k] then
19                     R[m][j][k] = maxGainSeq_{M,j} + R[m − 1][g][k − j]
20                     S[m][j][k] = {(M, j)} ∪ S[m − 1][g][k − j]
21                 else
22                     R[m][j][k] = R[m][j − 1][k]
23                     S[m][j][k] = S[m][j − 1][k]
```

erreichen kann, indem man nur die zusammengefügten Artikel mit Kennzahlen von eins bis m betrachtet und höchstens j atomare Artikel aus dem aktuellen m-ten zusammengefügten Artikel extrahiert.

Beweis. Wir werden dies durch ein Induktionsbeweis zeigen. Der Induktionsanfang ist die Aussage, dass $R[m][j][0]$ und $R[0][j][k]$ gleich null für alle m, j und k die korrekten Werte haben. Dies ist für $m = 0$ der Fall, da ein Artikel mit Kennzahl m kleiner-gleich null nicht existiert und damit keine Artikel vorhanden sind, aus denen man Extrahieren darf. Die Werte von j und k spielen hierbei keine Rolle. Für $k = 0$ Extraktionen insgesamt, muss der GainSeq ebenfalls null sein, da ohne Extraktionen kein Zitationswerte-Anstieg erreicht werden kann. Wieder gilt dies ungeachtet der anderen Argumente von R. Für die Induktionsbehauptung nehmen wir nun an, dass alle Ergebnisse aus $R[m][j][k]$, die nach der zuvor beschrieben Iterations-Reihenfolge beginnend von $R[0][0][0]$ bereits abgelaufen wurden, auch die korrekten Werte besitzen (Reihenfolge auch in Algorithmus 2 von den Schleifen ablesbar). Da in allen Fällen nur Werte aus R benötigt werden, die nach dieser Reihenfolge bereits durchgegangen wurden, müssen wir für den Induktionsschritt nur noch zeigen, dass für alle vier Fälle der rekursiven Formel auch die richtigen Werte ermittelt werden, wobei der erste Fall bereits vom Induktionsanfang abgedeckt wird.

Beim zweiten Fall stehen entweder insgesamt weniger Extraktionen k für die Gesamtfolge zur Verfügung, als speziell aus dem m-ten zusammengefügten Artikel extrahiert werden dürfen, oder man darf mehr aus ihm extrahieren, als es in ihm zu extrahieren gibt. In beiden dieser Situationen kann es im Vergleich zum maximalen GainSeq, mit einer Extraktion aus dem m-ten Artikel weniger, zu keinem Unterschied gekommen sein, weshalb man diesen Wert übernehmen kann.

Im dritten und vierten Fall ist die Abhängigkeit von $R[m][j][k]$ durch seine Vorgänger dadurch bestimmt, ob es durch *genau* j Extraktionen aus dem m-ten Artikel M (also durch $F_{M,j}$) zu einem größerem GainSeq mit k Extraktionen kommt, als es mit einer Extraktion aus M weniger der Fall ist. Der Wert für $R[m][j][k]$ ist damit das Maximum zwischen dem GainSeq der Teilfolge $F_{M,j}$ addiert mit dem bisher größtem GainSeq für $k - j$ Restextraktionen, die nicht aus M sind und zwischen dem größten GainSeq, der für genau eine Extraktion aus M weniger erreichbar ist. Fall 3 und 4 unterscheiden sich nur in der Hinsicht voneinander, wo dieser maximale GainSeq für eine Extraktion aus M weniger zu finden ist. Für den dritten Fall mit $j = 1$ wären keine Extraktion aus M erlaubt, weshalb er sich zur maximalen Anzahl an Extraktionen aus den Artikeln bis $m - 1$ nicht geändert haben kann und dieser Wert $R[m - 1][g][k]$ übernommen werden kann. Ansonsten ist er entsprechend gleich dem Wert von $R[m][j - 1][k]$. Damit wäre der Induktionsschluss gezeigt. □

5.3 Ermitteln der kürzesten maximierenden Folge F_{opt}

Der dritte und letzte Teilalgorithmus (Algorithmus 3) ermittelt nun aus den F^k die kürzeste den RealgainSeq maximierende Folge F_{opt}. Um diesen Teilalgorithmus zu begründen werden wir hauptsächlich die folgenden vier Aussagen Beweisen müssen:

Algorithm 3: Ermitteln der kürzesten maximierenden Folge

```
1  for M ∈ 𝒫 do
2      μ(M) = |citBy(M)|
3  topGet = g Artikel M ∈ 𝒫 mit größtem μ(M) sortiert nach aufsteigendem Wert
4  for k ∈ {0, ..., g} do
5      realgainSeq = 0
6      F^k = Extraktionen der Teilfolgen in S[|𝒫|][g][k] geordnet nach monoton fallenden
         GainInSeqs
7      for i ∈ {1, ..., k} do
8          if topGet(i) ≥ gainInSeq(i, F^k) then
9              break                                   // F^k ist kein Kandidat für F_opt
10         else
11             realgainSeq = realgainSeq + gainInSeq(i, F^k) − topGet(i)
12     if realgainSeq > RealGainSeqMax then
13         RealGainSeqMax = realgainSeq
14         F_opt = F^k
```

Zum einen ist zu zeigen, dass *die kürzeste Folge F_{opt} mit dem maximalen RealgainSeq eine der Folgen F^k mit maximalem GainSeq für genau k Extraktionen ist.* (I) Wenn wir dies bewiesen haben, müssen wir nur noch einen Weg finden F_{opt} unter diesen ausfindig zu machen.

Dazu zeigen wir zweitens, dass *wenn eine Folge mit monoton fallenden GainInSeqs eine Extraktion mit einem RealgainInSeq kleiner oder gleich null besitzt, dass dann alle darauffolgenden Extraktionen ebenfalls einen RealgainInSeq kleiner oder gleich null haben.* (II) Alle F^k für die eine solche Extraktion existiert sind damit von den potentiellen F_{opt} auszuschließen, da die Folge die diese Extraktion auslässt, aber alle anderen durchführt, mindesten den gleichen RealgainSeq besitzt und dabei auch noch kürzer ist.

Um aber zu bestimmen was für RealgainInSeqs die einzelnen Extraktionen in den Folgen haben nutzen wir aus, dass drittens *der RealgainInSeq der i-ten Extraktion einer Folge größer ist als null, genau dann wenn die Differenz seines GainInSeq mit der i-kleinsten Zitationswertung in den Top g der Startpartition $\mathcal{P}$ größer als null ist.* (III) Zuletzt müssen wir nur noch die RealgainSeqs der verbliebenen Kandidaten ermitteln und diese Vergleichen, um F_{opt} ausfindig zu machen.

Dazu verwenden wir die Tatsache, dass viertens *der RealGainSeq einer Folge der Länge k mit monoton fallenden GainInSeqs und ausschließlich RealgainInSeqs größer null, gleich der Differenz seines GainSeqs mit der Summe der Zitationswertungen der k am wenigsten zitierten Artikel in den Top g Artikeln der Partition $\mathcal{P}$ ist.* (IV) Diese vier Behauptungen werden wir in diesem Abschnitt formal beschreiben und beweisen. Dies tun wir aufgrund von Abhängigkeiten der Beweise

in der Reihenfolge (IV),(II),(I),(III).

Der Teilalgorithmus 3 nutzt aus, dass wegen Lemma 5 jede Folge in eine Folge mit monoton fallenden GainInSeqs umgeordnet werden kann, ohne seinen GainSeq oder RealGainSeq zu verändern. Für die folgenden Beweise definieren wir *OrderedByGain* als boolesche Hilfsfunktion für eine Folge $F = (f_1, \ldots, f_k)$ aus $\mathcal{F}$ und eine Partition $\mathcal{P}$, die nach dieser Eigenschaft prüft. In Formeln,

$$\text{orderedByGain}(F, \mathcal{P}) := \forall i \in \{1, \ldots, k-1\}.\text{gainInSeq}(\mathcal{P}, i, F) \geq \text{gainInSeq}(\mathcal{P}, i+1, F).$$

Um in eine solche Folge umzuordnen, muss man lediglich von Extraktion zu Extraktion schrittweise den atomaren Artikel zuerst extrahieren, der den größten Gain aufweist. Die GainInSeqs aller anderen atomaren Artikel der Folge werden lediglich kleiner oder bleiben gleich. Da wir die Teilfolgen $F_{M,i}$ im Teilalgorithmus 1 auf diese Weise extrahiert haben und sich Extraktionen aus verschiedenen zusammengefügten Artikeln nicht beeinflussen kann man die Artikel in den F^k nach den GainInSeq anordnen ohne die Extraaktionsreihenfolge innerhalb der Teilfolgen zu verletzen. Unter der Voraussetzung, dass eine Folge F der Länge k zuvor in diese Form umgeordnet wurde und, dass alle Extraktionen dieser umgeordneten Folge einen RealgainInSeq größer als null aufweisen, können wir nach Lemma 7b den RealgainSeq von F als Differenz des GainSeq von F und der Summe der Zitationswertungen der k-kleinsten Artikel in den *Top g der Startpartition* berechnen. Lemma 7a zeigt uns nämlich, dass auch genau der Artikel mit der k-kleinsten Zitationswertung bei der k-ten Extraktion extrahiert wird.

Lemma 7. *Gegeben eine Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$, ein $g \in \mathbb{N}$, sowie eine Folge $F \in \mathcal{F}$ mit $F := (f_1, \ldots, f_k)$, für die gilt*

i) $\text{orderedByGain}(F)$ *und*

ii) $\forall x \in \{1, \ldots, k\}.\, \text{realgainInSeq}(\mathcal{P}, x, F, g) > 0$,

dann folgt daraus

a) $\forall i \in \{1, \ldots, k\}.\, \text{topGet}(\mathcal{P}, g, i) = \text{topGet}(\mathcal{P}_{(f_1, \ldots, f_{i-1})}, g, 1)$ *und*

b) $\text{realgainSeq}(\mathcal{P}, F, g) = \text{gainSeq}(\mathcal{P}, F) - \sum_{i=1}^{k} \mu(\text{topGet}(\mathcal{P}, g, i))$.

Beweis. Wir beweisen zuerst Lemma 7a. Wegen Lemma 4a und Vorbedingung ii) wissen wir, dass alle Teilartikel, die bei der i-te Extraktionen innerhalb der Extraktionsfolge entstehen, auch in den Top g der dabei entstehenden Partition $\mathcal{P}_{(f_1, \ldots, f_i)}$ enthalten sind. In Formeln,

$$\forall i \in \{1, \ldots, k\}.M_{f_i, \mathcal{P}_{(f_1, \ldots, f_{i-1})}} \setminus \{f_i\} \in \text{top}(\mathcal{P}_{(f_1, \ldots, f_i)}, g) \wedge \{f_i\} \in \text{top}(\mathcal{P}_{(f_1, \ldots, f_i)}, g). \quad (1)$$

Da (1) gilt können wir schlussfolgern, dass bei jeder Extraktion f_i von F der Artikel mit der zu dem Zeitpunkt geringsten Zitationswertung in den Top g, also $\text{topGet}(\mathcal{P}_{(f_1, \ldots, f_{i-1})}, g, 1)$, aus

den größten g verdrängt wird. Sei v ein beliebiger Artikel $v \in \bigcup_{i=1}^{k} \text{topGet}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, g, 1)$, der im i-ten Extraktionsschritt aus den Top g verdrängt wird. Wenn wir zeigen können, dass es sich hierbei nur um Artikel handelt, die schon in der Startpartition $\mathcal{P}$ in Top g waren, also in $\text{top}(\mathcal{P}, g)$ und diese nicht durch eine Extraktion von F entstandene Teilartikel sind, dann wäre der Beweis zu ende, da die k-kleinsten Artikel zuerst verdrängt werden müssen. Artikel $a \in \mathcal{P} \wedge a \notin \text{top}(\mathcal{P}, g)$ sind wegen (1) von den verdrängten Artikel v auszuschließen, da bei jeder Extraktion stets zwei neue Artikel zu den momentanen Top g hinzukommen. Von außerhalb der Top g kann also kein Artikel hinein rücken. Zu zeigen ist also nur noch, dass es sich bei den verdrängten v nicht um neu entstandene Teilartikel handelt. Da wegen orderedByGain(F) und Lemma 2b alle durch die Extraktionsfolge entstandenen Teilartikel eine Zitationswertung besitzen die größer oder gleich dem GainInSeq aller darauffolgenden Extraktionen sind, ist es nicht möglich, dass eines dieser Teilartikel aus den Top g verdrängt wird, ohne dass der Realgain kleiner-gleich null ist. Dies folgt nämlich direkt aus Lemma 4b und gäbe es einen Widerspruch mit Vorbedingung ii). Damit müssen alle verdrängten Artikel v auch die mit den k-kleinsten Zitationswertungen der Top g Artikel der Startpartition sein.

Der Beweis von Lemma 7b folgt aus Lemma 7a, durch

$$\text{realgainSeq}(\mathcal{P}, (f_1, \ldots, f_k), g) \underset{\text{Def. realgainSeq}}{=} \sum_{i=1}^{k} \text{realgain}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, f_i, g) \underset{\text{Lemma 3e und (1)}}{=}$$

$$\text{gainSeq}(\mathcal{P}, F) - \sum_{i=1}^{k} \mu(\text{topGet}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, g, 1)) \underset{\text{Lemma 5a}}{=} \text{gainSeq}(\mathcal{P}, F) - \sum_{i=1}^{k} \mu(\text{topGet}(\mathcal{P}, g, i)).$$

□

Die Einschränkung von Lemma 7b, dass wir den RealgainSeq auf diese Weise nur berechnen können, wenn die einzelnen RealgainInSeqs größer als null sind stört uns nicht, da wie wir später zeigen werden, Folgen die nach monoton fallenden GainInSeqs angeordnet wurden, für die dies aber nicht gilt, auch keine Kandidaten für die gesuchte kürzeste maximierende Folge darstellen. Zuerst müssen wir aber im nun folgenden Lemma 8 beweisen, dass falls eine Extraktion, in einer Folge die nach monoton sinkenden GainInSeq angeordnet wurde, einen negativen RealgainInSeq aufweist, dass dies auch für alle darauffolgenden Extraktionen in der Folge gilt. Diese Erkenntnis wird uns später von Nutzen sein.

Lemma 8. *Gegeben eine Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$, ein $g \in \mathbb{N}$, ein $x \in \mathbb{N}$, sowie eine Folge $F \in \mathcal{F}$ mit $F := (f_1, \ldots, f_k)$, für die gilt*

i) orderedByGain(F) *und*

ii) realgainInSeq$(\mathcal{P}, x, F, g) \leq 0$,

dann folgt daraus

$$\forall y \in \{x+1, \ldots, k\}.\, \text{realgainInSeq}(\mathcal{P}, y, F, g) \leq 0.$$

Beweis. Wir beweisen dies durch eine Induktion. Sei f_x der x-te atomare Artikel in einer Folge $F := (f_1, \ldots, f_k)$ von Extraktionen für die gilt orderedByGain(F) und realgainInSeq($\mathcal{P}, x, F, g) \leq 0$. Dies bildet unseren Induktionsanfang. Die zu zeigende Induktionsbehauptung ist damit, dass für alle beliebigen $i \in \{x, \ldots, k\}$ gilt

$$\text{realgainInSeq}(\mathcal{P}, i, F, g) \leq 0 \rightarrow \text{realgainInSeq}(\mathcal{P}, i+1, F, g) \leq 0.$$

Nun folgt der Induktionsschluss. Wir nehmen im Folgendem an, dass die Prämisse der Induktionsbehauptung realgainInSeq($\mathcal{P}, i, F, g) \leq 0$ wahr ist. Wegen Lemma 4b gilt für f_{i+1}, dass

$$\mu(\text{topGet}(\mathcal{P}_{(f_1,\ldots,f_i)}, g, 1)) \geq \text{gainInSeq}(\mathcal{P}, i+1, F) \leftrightarrow \text{realgainInSeq}(\mathcal{P}, i+1, F, g) \leq 0, \quad (1)$$

ist. Wegen Lemma 4b und realgainInSeq($\mathcal{P}, i, F, g) \leq 0$ ist uns auch bekannt, dass für die vorherige Extraktion f_i

$$\mu(\text{topGet}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, g, 1)) \geq \text{gainInSeq}(\mathcal{P}, i, F)$$

gilt. Da also der GainInSeq der i-ten Extraktion aus F kleiner-gleich der Zitationswertung des kleinsten Artikels in den Top g vor der Extraktion ist, muss dies auch für die restlichen Artikel in den Top g gelten, also

$$\forall v \in \text{top}(\mathcal{P}_{(f_1,\ldots,f_{i-1})}, g).\mu(v) \geq \text{gainInSeq}(\mathcal{P}, i, F).$$

Durch Lemma 2b wissen wir, dass beide durch die Extraktion f_i entstandenen Teilartikel $\{f_i\}$ und $M_{f_i,\mathcal{P}_{(f_1,\ldots,f_{i-1})}} \setminus \{f_i\}$ jeweils eine Zitationswertung haben, die größer oder gleich dem GainInSeq der Extraktion ist, durch die sie entstanden sind. Falls der Artikel $M_{f_i,\mathcal{P}_{(f_1,\ldots,f_{i-1})}}$ aus dem Extrahiert wurde also in top($\mathcal{P}_{(f_1,\ldots,f_{i-1})}, g$) enthalten war, dann wird er in top($\mathcal{P}_{(f_1,\ldots,f_i)}, g$) von einem Teilartikel ersetzt der eine Zitationswertung besitzt, die mindestens gainInSeq($\mathcal{P}, i, F$) entspricht. Falls er nicht darin enthalten war bleiben die Top g nach der Extraktion gleich. Damit muss für top($\mathcal{P}_{(f_1,\ldots,f_i)}, g$) genau wie auch schon zuvor für top($\mathcal{P}_{(f_1,\ldots,f_{i-1})}, g$) gelten, dass die Zitationswertungen der darin enthaltenen Artikel mindestens so groß sind wie der GainInSeq der i-ten Extraktion, also

$$\forall v \in \text{top}(\mathcal{P}_{(f_1,\ldots,f_i)}, g).\mu(v) \geq \text{gainInSeq}(\mathcal{P}, i, F).$$

Daraus folgt wegen den monoton sinkenden GainInSeqs von F, dass

$$\forall v \in \text{top}(\mathcal{P}_{(f_1,\ldots,f_i)}, g).\mu(v) \geq \text{gainInSeq}(\mathcal{P}, i, F) \geq \text{gainInSeq}(\mathcal{P}, i+1, F).$$

Damit muss die Induktionsbehauptung wegen (1) korrekt sein. □

Nun werden wir uns der Frage widmen, wieso F_{opt} in einem der F^k zu finden ist. Wie bereits reichlich erwähnt, handelt es sich bei den F^k um Folgen von Extraktionen der Länge k mit den größten erreichbaren GainSeqs für alle (abgesehen von der leeren Folge) relevanten Folgenlängen $k \in \{1, \ldots, g\}$, mit monoton sinkenden GainInSeqs. Alle Folgen F die länger als g sind würden

mehr Teilartikel erzeugen, als in den Top g reinpassen, wodurch diese verdrängt werden. Da Teilartikel nach Lemma 1a immer kleinere Zitationswertungen besitzen als ihre Ursprungsartikel, die gleichzeitig größer sind als die GainInSeqs ihrer Extraktion und da orderedByGain$(F, \mathcal{P})$ gilt, würden diese Folgen nach Lemma 4b spätesten ab der $g+1$-ten Extraktion nur RealgainInSeqs kleiner oder gleich null aufweisen, weshalb diese keine Kandidaten für F_{opt} darstellen. F_{opt} muss damit eine Folge mit einer Länge von 0 bis g sein.

Sei F die Folge mit dem gesuchten maximalen RealgaInSeq mit dabei minimaler Länge und sei E eine Folge mit dem maximalen GainSeq der mit k Extraktionen erreicht werden kann, wobei F ebenfalls die Länge k hat und beide nach monoton sinkenden GainInSeqs angeordnet wurden, dann hat nach Lemma 9a E nur RealgainInSeqs größer als null. Daraus folgt dann in Lemma 9b, dass E auch eine den RealGainSeq maximierende Folge mit dabei minimaler Länge ist. Damit muss F_{opt} eines der F^k sein.

Lemma 9. *Gegeben eine Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$, ein $g \in \mathbb{N}$, ein $k \in \mathbb{N}$ sowie zwei Folgen $F, E \in \mathcal{F}$ mit $F := (f_1, \ldots, f_k)$ und $E := (e_1, \ldots, e_k)$, der selben Länge k, für die gilt*

i) orderedByGain(F)*,*

ii) orderedByGain(E)*,*

iii) *F ist maximierend bezüglich des RealgainSeq mit dabei minimaler Länge, das heißt:*
$\neg\exists H \in \mathcal{F}.\, \text{realgainSeq}(\mathcal{P}, H, g) > \text{realgainSeq}(\mathcal{P}, F, g) \wedge$
$\neg\exists H \in \mathcal{F}.\, \text{realgainSeq}(\mathcal{P}, H, g) = \text{realgainSeq}(\mathcal{P}, F, g) \wedge \text{Länge}(H) < \text{Länge}(F)$ *und*

iv) *E hat den größtmöglichen GainSeq für genau k Extraktionen, das heißt:*
$\neg\exists H \in \mathcal{F}.\, \text{gainSeq}(\mathcal{P}, H, g) > \text{gainSeq}(\mathcal{P}, E, g) \wedge \text{Länge}(H) = \text{Länge}(E)$*,*

dann folgt daraus

a) $\forall i \in \{1, \ldots, k\}.\, \text{realgainInSeq}(\mathcal{P}, i, E, g) > 0$ *und*

b) *E ist ebenfalls maximierend bezüglich des RealgainSeq mit dabei minimaler Länge, das heißt:*
$\neg\exists H \in \mathcal{F}.\, \text{realgainSeq}(\mathcal{P}, H, g) > \text{realgainSeq}(\mathcal{P}, E, g) \wedge$
$\neg\exists H \in \mathcal{F}.\, \text{realgainSeq}(\mathcal{P}, H, g) = \text{realgainSeq}(\mathcal{P}, E, g) \wedge \text{Länge}(H) < \text{Länge}(E)$

Beweis. Wir beginnen zuerst mit dem Beweis von Lemma 9a. Weil orderedByGain(F) gilt, bedeutet das, dass auch

$$\forall i \in \{1, \ldots, k\}.\, \text{realgainInSeq}(\mathcal{P}, i, F, g) > 0$$

gilt, was wir mit einem Widerspruchsbeweis zeigen können. Angenommen es gäbe eine Extraktion für die dies nicht gilt, also

$$\textit{Annahme 1: } \exists i \in \{1, \ldots, k\}\, \text{realgainInSeq}(\mathcal{P}, i, F, g) \leq 0$$

ist, dann wären nach Lemma 8 auch die RealgainInSeqs aller darauffolgenden Extraktionen kleiner-gleich null. Damit gäbe es eine kürzere Subfolge von F mit größer oder gleichem RealgainSeq, die mit der Vorbedingung iii) in einem Widerspruch stehen würde. Annahme 1 wäre damit nicht möglich. Wir beweisen nun die Aussage von Lemma 9a mit einem weiteren Widerspruchsbeweis, indem wir das selbe auch für E annehmen, also

$$\textit{Annahme 2: } \exists i \in \{1, \ldots, k\}.\, \text{realgainInSeq}(\mathcal{P}, i, E, g) \leq 0.$$

Weil orderedByGain(E) gilt und die Extraktionen von E damit immer kleiner werdende Gains aufweisen, müssen alle Extraktionen nach der ersten Extraktion x mit RealgainInSeq ≤ 0 wegen Lemma 8 ebenfalls ein RealgainInSeq ≤ 0 besitzen. Dieses x ist so definiert, dass für es gilt:

$$\forall i \in \{x, \ldots, k\}.\, \text{realgainInSeq}(\mathcal{P}, i, E, g) \leq 0 \wedge \forall w \in \{1, \ldots, x-1\}.\, \text{realgainInSeq}(\mathcal{P}, w, E, g) > 0 \tag{1}$$

Wegen Lemma 4b folgt daraus für alle i-ten Extraktionen ab dieser x-ten Extraktion in E, dass in der Startpartition $\mathcal{P}$ alle Zitationswertungen der i-kleinsten Artikel aus den Top g auch größer oder gleich den i-ten GainInSeqs sind. In Formeln,

$$\forall i \in \{x, \ldots, k\}.\, \text{gainInSeq}(\mathcal{P}, i, E) \leq \mu(\text{topGet}(\mathcal{P}, g, i)). \tag{2}$$

Für F folgt durch Lemma 4b hingegen das Gegenteil für alle seine Extraktionen, mit

$$\forall i \in \{1, \ldots, k\}.\, \text{gainInSeq}(\mathcal{P}, i, F) > \mu(\text{topGet}(\mathcal{P}, g, i)) \underset{\text{Lemma 7}a}{=} \text{topGet}(\mathcal{P}_{(e_1,\ldots,e_{i-1})}, g, 1). \tag{3}$$

Und somit, dass alle i-ten GainInSeqs von F ab der x-ten Extraktion aufwärts größer sind als die i-ten von E:

$$\forall i \in \{x, \ldots, k\}.\, \text{gainInSeq}(\mathcal{P}, i, F) \underset{(3)}{>} \mu(\text{topGet}(\mathcal{P}, g, i)) \underset{(2)}{\geq} \text{gainInSeq}(\mathcal{P}, i, E) \tag{4}$$

Da aber insgesamt wegen Vorbedingung iv) gainSeq($\mathcal{P}, E$) > gainSeq($\mathcal{P}, F$) gilt, muss E von 1 bis $x-1$ „kompensieren“, dass es von x bis k kleinere GainInSeqs als F hat, indem

$$\sum_{i=1}^{x-1} \text{gainInSeq}(\mathcal{P}, i, E) \geq \sum_{i=1}^{x-1} \text{gainInSeq}(\mathcal{P}, i, F) + \text{Diff} \tag{5}$$

ist. Wobei Diff die Differenz der Teil-GainSeqs ab x darstellt, die es von x aufwärts zu kompensieren gilt. Diff muss dabei ein Wert größer als null sein. In Formelschreibweise wäre dies also

$$\text{Diff} = \sum_{i=x}^{k} \text{gainInSeq}(\mathcal{P}, i, F) - \text{gainInSeq}(\mathcal{P}, i, E).$$

Und da (4) gilt, gilt damit auch

$$\text{Diff} \geq \sum_{i=x}^{k} \text{gainInSeq}(\mathcal{P}, i, F) - \sum_{i=x}^{k} \mu(\text{topGet}(\mathcal{P}, g, i)). \tag{6}$$

Wegen Lemma 7b und da laut (1) in E von Extraktion 1 bis $x-1$ die RealgainInSeqs größer null sind, lassen sich dann die Realgains der beiden Folgen in diesem Bereich in Verhältnis setzen, indem

$$\begin{aligned}\sum_{i=1}^{x-1}\text{realgainInSeq}(\mathcal{P},i,E,g) &\underset{\text{Lemma 7}b}{=} \sum_{i=1}^{x-1}\text{gainInSeq}(\mathcal{P},i,E) - \sum_{i=1}^{x-1}\mu(\text{topGet}(\mathcal{P},g,i)) \\ &\underset{(5)}{\geq} \sum_{i=1}^{x-1}\text{gainInSeq}(\mathcal{P},i,F) + \text{Diff} - \sum_{i=1}^{x-1}\mu(\text{topGet}(\mathcal{P},g,i)) \qquad (7)\\ &\underset{\text{Lemma 7}b}{=} \sum_{i=1}^{x-1}\text{realgainInSeq}(\mathcal{P},i,F) + \text{Diff}\end{aligned}$$

ist. Die lässt sich verkürzen zu

$$\text{realgainSeq}(\mathcal{P},(e_1,\ldots,e_{x-1}),g) = \sum_{i=1}^{x-1}\text{realgainInSeq}(\mathcal{P},i,E,g) \underset{(7)}{\geq} \sum_{i=1}^{x-1}\text{realgainInSeq}(\mathcal{P},i,F)+\text{Diff}. \qquad (8)$$

Der Realgain von F dagegen lässt sich ebenfalls durch Lemma 7b umschreiben als

$$\text{realgainSeq}(\mathcal{P},F,g) = \sum_{i=1}^{x-1}\text{realgainInSeq}(\mathcal{P},i,F)+(\sum_{i=x}^{k}\text{gainInSeq}(\mathcal{P},i,F)-\sum_{i=x}^{k}\mu(\text{topGet}(\mathcal{P},g,i))). \qquad (9)$$

und daraus folgt letztendlich

$$\text{realgainSeq}(\mathcal{P},(e_1,\ldots,e_{x-1}),g) \underset{(8)}{=} \sum_{i=1}^{x-1}\text{realgainInSeq}(\mathcal{P},i,F) + \text{Diff} \underset{(9\ \&\ 6)}{\geq} \text{realgainSeq}(\mathcal{P},F,g).$$

Dies ist jedoch ein Widerspruch zu iii). Damit ist Annahme 2 falsch und Lemma 9a muss gelten. Lemma 9b folgt dann, da E und F damit beide nur Extraktionen mit Realgains größer null besitzen und Lemma 7b damit auf sie anwendbar ist um ihre RealgainSeqs zu berechnen. Die RealgainSeqs beider Folgen lassen sich damit durch die Differenz ihrer GainSeqs und der gleichen Summe der Zitationswertungen der k kleinsten Artikeln aus den Top g der Startpartition berechnen. Da E und F die selbe Anzahl an Extraktionen k besitzen und der GainSeq von E für k Extraktionen maximal ist, muss E damit eines der möglichen F sein. □

Nun brauchen wir nur noch eine einfache Methode anhand der GainInSeqs zu prüfen, ob eine einzelne Extraktion einen RealgainInSeq größer null oder kleiner-gleich null besitzt. Dazu dient das letzte Lemma 10. Es hat eine ähnliche Aussage wie Lemma 4b bezieht sich jedoch auf eine Extraktion *innerhalb einer Extraktionsfolge*. Es sagt aus, dass der RealgainInSeq der i-ten Extraktion einer Folge größer ist als null, genau dann wenn die Differenz seines GainInSeq mit der i-kleinsten Zitationswertung in den Top g der Startpartition $\mathcal{P}$ größer als null ist.

Lemma 10. *Gegeben eine Partition $\mathcal{P}$ von $W \subseteq V$ bezüglich eines Zitationsgraphen $D = (V, Z)$, ein $g \in \mathbb{N}$, ein $x \in \mathbb{N}$, sowie eine Folge $F \in \mathcal{F}$ mit $F := (f_1, \ldots, f_k)$, für die gilt* orderedByGain(F), *dann folgt daraus*

$$\forall x \in \{1, \ldots, k\}.\, \text{realgainInSeq}(\mathcal{P}, x, F, g) > 0 \Leftrightarrow \text{gainInSeq}(\mathcal{P}, x, F) > \text{topGet}(\mathcal{P}, g, x)$$

Beweis. Wir beweisen zuerst die Hinrichtung. Falls der RealGainInSeq größer als null ist gilt dies wegen Lemma 8 auch für alle vorherigen Extraktionen. Damit wissen wir durch Lemma 7a auch, dass der Artikel mit der x-kleinsten Zitationswertung aus den Top g der Startpartition topGet($\mathcal{P}, g, x$) auch der kleinste Artikel den Partition $\mathcal{P}\{1, \ldots, x-1\}$ direkt vor der Extraktion x darstellt. Damit folgt aus der Prämisse und Lemma 4b die Konklusion.

Als nächstes beweisen wir die Rückrichtung. Da wegen orderedByGain(F) alle vorherigen Extraktionen einen größeren GainInSeq als die aktuelle Extraktion x haben, wissen wir durch Lemma 2b, das alle Zitationswertungen der Teilartikel die in den vorherigen Extraktionen entstanden sind größer oder gleich als topGet($\mathcal{P}, g, x$) bis topGet($\mathcal{P}, g, 1$) sind. Wegen Lemma 2a ist es damit nicht möglich, dass die zusammengefügten Artikel aus denen extrahiert wurde die x kleinsten Artikel aus den Top g der Startpartition sind, da Teilartikel im allgemeinen nicht eine größere Zitationswertung haben können, als der zusammengefügten Artikel aus dem extrahiert wurde. Damit müssen alle Teilartikel der ersten x Extraktionen es in die top($\mathcal{P}_{(f_1,\ldots,f_x)}, g$) geschafft haben, wobei sie die topGet($\mathcal{P}, g, 1$) bis topGet($\mathcal{P}, g, x$) schrittweise verdrängt haben. Daraus folgt aus der Prämisse und Lemma 4b, dass alle Extraktionen bis einschließlich dem x-ten einen positiven RealgainInSeq haben. □

In den Zeilen 1 bis 3 des Teilalgorithmus 3 (Algorithmus 3) werden die g Artikel mit den größten Zitationswertungen top($\mathcal{P}, g$) ermittelt und sortiert in einer Liste gespeichert. Wenn wir nun die kürzeste Folge mit dem maximalen RealgainSeq F_{opt} ermitteln wollen, müssen wir lediglich noch für alle möglichen Folgenlängen $k \in \{1, \ldots, g\}$ die Folgen F^k mit den maximalen GainSeqs für genau k Extraktionen mit monoton sinkenden GainInSeqs ermitteln. Es sollte beachtet werden, dass die in Zeile 6 betrachteten Einträge aus S, wie in Teilabschnitt 5.2 beschrieben, die Mengen der benötigten Teilfolgen $F_{M,i}$ darstellen, um die F^k zusammenzusetzen. Diese wiederum enthalten die in Zeile 24 aus Teilalgorithmus 1 (Algorithmus 1) erzeugten Zweier-Tupel, die als ersten Eintrag den zu extrahierenden Artikel und als zweiten dessen GainInSeq enthalten. Wichtig ist, dass diese Teilfolgen $F_{M,i}$ in Algorithmus 1 nach monoton sinkenden GainInSeqs ermittelt wurden, weshalb ihre GainInSeqs in F^k unverändert bleiben, da dieser ebenfalls nach monoton sinkenden GainInSeqs angeordnet wird.

Danach bilden wir für alle F^k die Differenzen gainInSeq($\mathcal{P}, i, F^k$) − μ(topGet($\mathcal{P}, g, i$)) für alle $i \in \{1, ..., k\}$. Ist eine dieser Differenzen kleiner oder gleich null, hat dieses F^k nach Lemma 10 RealgainInSeqs die kleiner oder gleich null sind. Damit kann die maximierende Folge nach Lemma 9a nicht in den Folgen der Länge k zu finden sein. Für die restlichen F^k, in denen alle diese Differenzen größer als null sind, bilden wir die Summe dieser Differenzen und erhalten damit nach Lemma 7b den RealgainSeq von F^k. Die kürzeste Folge F^k mit dem größten RealgainSeq

ist dann die gesuchte maximierende Folge F_{opt}. Für diese muss man lediglich noch prüfen, ob sie den benötigten g-Index erreicht hat. Ist dies der Fall gibt man sie aus, ansonsten gibt man $\perp$ aus.

5.4 Laufzeitanalyse

Theorem 2. *g-Index-Manipulation durch Artikel-Extraktion ist lösbar in* $\mathcal{O}(|\mathcal{P}| \cdot 2^{\min(g,M_{\max})} \cdot M_{\max} \cdot |E| + |\mathcal{P}| \cdot g^2 + |\mathcal{P}|^2 + g^3)$ *Zeit, für eine Partition* $\mathcal{P}$ *von* $W \subseteq V$ *bezüglich eines Zitationsgraphen* $D = (V, Z)$*, sein entsprechendes* $M_{\max} = \max_{M \in \mathcal{P}}(|M|)$ *und ein* $g \in \mathbb{N}$*, der den zu erreichenden g-Index angibt.*

Beweis. Von allen drei Teilalgorithmen hat der erste den größten Worst-Case-Aufwand. Seine Laufzeit wird von der Schleife von Zeile 12 bis 35 und seinen inneren Schleifen dominiert, in der für alle zusammengefügte Artikel $M \in \mathcal{P}$ alle Kombinationen T an Teilmengen von atomaren Artikeln aus M die kleiner als g sind durchgegangen werden. Sei $M_{\max} = \max_{M \in \mathcal{P}}(|M|)$, dann beträgt der Aufwand der Schleife damit $\mathcal{O}(|\mathcal{P}| \cdot 2^{\min(g,M_{\max})} \cdot x)$, wobei x den Aufwand der Operationen innerhalb der Schleifendurchläufe der verschiedenen T beschreibt. Der Aufwand lässt sich damit durch $\mathcal{O}(|\mathcal{P}| \cdot 2^{M_{\max}} \cdot x)$ und $\mathcal{O}(|\mathcal{P}| \cdot 2^{g} \cdot x)$ einschränken. Nun ermitteln wir den Aufwand von x der hauptsächlich durch die For-Schleife von den Zeilen 21 bis 32 geprägt wird. Hier werden für alle atomare Artikel in T alle atomaren Artikel durchgegangen, die in M zitieren, welche im Worst-Case alle Kanten aus E im Graph G sind. Damit lässt sich x durch $M_{\max} \cdot |E|$ einschränken und der erste Teilalgorithmus hat damit insgesamt einen Aufwand von $\mathcal{O}(|\mathcal{P}| \cdot 2^{\min(g,M_{\max})} \cdot M_{\max} \cdot |E|)$.

Der zweite Teilalgorithmus besteht aus drei Schleifen von denen die Erste $|\mathcal{P}|$ Durchläufe besitzt und die anderen beiden g viele. Da die Operationen innerhalb der Schleifen nur simple Lese-,Schreib- und Vergleichsoperationen sind, lässt sich der Gesamtaufwand dieses Teilalgorithmus auf $\mathcal{O}(|\mathcal{P}| \cdot g^2)$ einschränken.

Der dritte Teilalgorithmus enthält zwei Sortieroperationen. Die Erste lässt sich durch einen Aufwand von $\mathcal{O}(|\mathcal{P}|^2)$ beschränken, da $|\mathcal{P}|$ Elemente sortiert werden. Die Zweite liegt innerhalb einer Schleife von g Durchläufen, sortiert weniger als g Elemente und hat damit insgesamt einen Aufwand von $\mathcal{O}(g^3)$. Der letzte Teilalgorithmus hat damit insgesamt einen Aufwand von $\mathcal{O}(|\mathcal{P}|^2 + g^3)$. Kombiniert man die Aufwände der Teilalgorithmen erhält man $\mathcal{O}(|\mathcal{P}| \cdot 2^{\min(g,M_{\max})} \cdot M_{\max} \cdot |E| + |\mathcal{P}| \cdot g^2 + |\mathcal{P}|^2 + g^3)$. □

6 Experimente

In diesem Abschnitt werden wir durch ein möglichst praxisnahes Experiment den Rechenaufwand und auch das Potential zur Verbesserung des g-Indexes unseres entwickelten Algorithmus untersuchen.

6.1 Daten und Implementierung

In unseren Experimenten lassen wir den Algorithmus für mehrere Autoren, mit ihren jeweiligen Zitationsgraphen, für verschiedene zu erreichende g-Indizes durchlaufen. Wir beginnen dabei bei einem g-Index von eins und inkrementieren g solange bis keine Verbesserungen mehr möglich sind. Dies ist der Fall, wenn für das aktuelle g, trotz der Erhöhung der Zitationen in den Top g durch den Algorithmus, der g-Index nicht erreicht wurde und gleichzeitig der Anstieg an zusätzlichen hierfür benötigten Zitationen von g zu $g+1$ in den Top-Artikeln (also $(g+1)^2 - g^2$) größer als die Summe der Gains aller atomaren Artikel des Autors ist.

Testdaten Es wurden die selben Test-Daten verwendet, die auch schon für die Tests der Arbeit von Bevern et al. [3] benutzt wurden. Hierbei handelt es sich um Daten von 28 gecrawlten Google Scholar Profilen. 14 dieser sind von Teilnehmern der Konferenz IJCAI'13, die früh in ihrer Karriere sind und für die sich deshalb eine Index-Manipulierung eher anbieten würde, als für bereits etablierte Forscher. Sie besaßen zum Zeitpunkt der Datenerstellung einen h-Index von 8 bis 20 und zwischen 100 und 1000 Zitate, waren in den letzten 5 bis 10 Jahren aktiv und haben keine Anstellung als Professor. Die restlichen Autoren sind von der „AI's 10 to Watch"-Auswahl von ausgezeichneten jungen Forschern. Sechs dieser sind von der 2011-er- und acht von der 2013-er-Ausgabe [1, 10].

Erstellung der Partition In den Daten fehlen jedoch die Informationen darüber, welche Artikel zusammengefügt wurden, weshalb wir dies möglichst realitätsgetreu simulieren mussten. Hierfür haben wir die Ähnlichkeit der Titel von Artikeln betrachtet und anhand eines Ähnlichkeitsmaßes bewertet. Dabei wurden die atomaren Artikel eines Autors, die als ähnlich bewertet wurden, durch eine Kante in einem ungerichteten Ähnlichkeits-Graph verbunden. Aus den Cliquen dieser Ähnlichkeits-Graphen haben wir dann unsere Test-Partitionen nach zwei bestimmten Methoden erstellt.

Bei der ersten Methode handelt es sich um einen Greedy-Algorithmus für ein M_{max} maximaler Größe, der ausgehend von der Menge aller maximalen Cliquen schrittweise stets den größten verbleibenden Artikel-Kandidaten für die Partition auswählt. Aus den übrigen Kandidaten wird dann die Schnittmenge mit dieser Clique entfernt. Diese Prozedur wird wiederholt bis keine Kandidaten mehr übrig sind. Der zweite Partitionierungsansatz unterscheidet sich lediglich dadurch vom Ersten, dass in jedem Schritt immer die kleinste der verbleibenden Cliquen ausgewählt wurde. Bei beiden Varianten wurden entstehende leere Cliquen aus der Menge der Kandidaten verworfen. Es sollte angemerkt werden, dass keine der beiden Partitionierungsmethoden auch die Partitionen liefern, die die größten g-Index-Verbesserungen ermöglichen, was auch nicht zwangsläufig unsere Motivation ist. Vielmehr sollen sie uns einigermaßen realistische Anwendungsbeispiele liefern an denen wir unseren Algorithmus austesten können.

Bei dem verwendeten Ähnlichkeitsmaß handelt es sich um eines aus der Arbeit von Bevern et al. [3], dessen Strenge durch die Ähnlichkeitsschranke $t \in [0, 1]$ beschrieben wird. Zwei Artikel u und v werden dabei als ähnlich erachtet, falls $|T(u) \cap T(v)| \geq t \cdot |T(u) \cup T(v)|$, wobei $T(u)$ die Anzahl der Wörter im Titel eines Artikels u ist [3]. Wir haben den kompletten beschriebenen Ablauf der Experimente *für alle Ähnlichkeitsschranken* von $t = 0,4$ bis $t = 0,9$ (in $0,1$-er Schritten) separat durchlaufen. Da für $t \leq 0,3$ relativ unähnliche Titel bereits als ähnlich betrachtet werden, haben wir diese nicht begutachtet. Es sollte beachtet werden, dass die Tatsache ob Verbesserungen möglich sind sehr stark von der Konstellation des Ähnlichkeits-Graphen und den daraus folgenden ausgewählten Cliquen und zusammengefügten Artikeln abhängig ist. So kann es für die verschiedenen Ähnlichkeitsschranken t zu mehr oder weniger unterschiedlichen Ergebnissen kommen, wenn nicht auch die atomaren Artikel, die untereinander viele zitierende Artikel teilen, im selben zusammengefügten Artikel landen.

Testumgebung Beim verwendeten System handelt es sich um ein Gerät mit einem Windows 7 Betriebssystem. Es hat ein 16GB DDR3 Arbeitsspeicher und der Prozessor ist ein Intel Core i5-2430M mit zwei Kernen mit je 2,4GHz. Das Programm wurde in Python geschrieben, wobei als essentielle Bibliothek von NetworkX Gebrauch gemacht wurde. Es wurde keine Parallelisierung in die Implementierung des Programms eingebaut.

6.2 Ergebnisse

Verbesserungen des g-Index waren bei der Partitionierung nach maximalem M_{max} für fünf Autoren zu beobachten. Für einen dieser Autoren konnte sogar der g-Index um zwei erhöht werden, allerdings nur für einen Ähnlichkeits-Graph mit $t = 0,4$. Die Laufzeiten der einzelnen Durchläufe des Algorithmus lagen in den meisten Fällen unter einer Sekunde und im Maximum bei knapp unter fünf Sekunden. Dies ist dadurch zu begründen, dass die zusammengefügten Artikel relativ klein sind. Das gilt vor allem für kleinere und damit weniger strengere Ähnlichkeits-Schranken t. Für $t = 3.0$, für den entsprechend relativ unähnliche Artikel als ähnlich bewertet wurden, waren Laufzeiten von bis zu knapp einer halben Minute zu beobachten, was durch die weitaus größeren zusammengefügten Artikel zu begründen ist. Diese haben wir jedoch, wie bereits erwähnt, nicht in unsere Ergebnisse gewertet. Überdurchschnittlich hoch war die Laufzeit wie zu erwarten bei gleichzeitig großen g und M_{max}. Die Ergebnisse für die Partitionierung der atomaren Artikel nach Auswahl der kleinsten Cliquen zuerst und damit auch einem kleinerem M_{max} sehen relativ ähnlich aus. Allerdings waren hier die Verbesserungen der g-Indexe der Autoren, wie auch anzunehmen war, geringer in der Zahl. Drei Autoren konnten jeweils ihren g-Index um eins steigern.

Im Folgendem geben wir unsere Ergebnisse in Tabellenform wieder. Es sollte angemerkt werden, dass der Algorithmus nach keinen dieser beiden Werte optimiert hat, sondern stattdessen lediglich die Anzahl der Zitate in den Top g für die gegebene Partition maximiert. So ist es möglich, dass der selbe g-Index mit weniger Extraktionen für die gleiche Eingabe erreichbar ist,

als in unseren Ergebnissen ermittelt wurde.

Die Spalte „#Extr.“ steht für die Anzahl der Extraktionen der vom Algorithmus ausgegebenen Folge. Die Spalte „#versch. zus. Art.“ beschreibt die Anzahl der verschiedenen zusammengefügten Artikel, aus denen dabei extrahiert wird.

Wir beginnen zuerst mit den Ergebnissen für die Partitionierung der atomaren Artikel nach maximalem M_{max}:

Verbesserungen für $t = 0,4$

Autor	g-Index	max. Verb.	Laufzeit	$\|E\|$	$\|\mathcal{P}\|$	M_{max}	#Extr.	#versch. zus. Art.
A	14	1	0,032s	245	20	3	2	2
B	28	1	0,201s	857	32	5	4	3
C	26	2	3,269s	833	42	11	5	1
E	21	1	0,132s	496	22	3	2	2
D	23	1	0,089s	590	22	4	3	3

Verbesserungen für $t = 0,5$

Autor	g-Index	max. Verb.	Laufzeit	$\|E\|$	$\|\mathcal{P}\|$	M_{max}	#Extr.	#versch. zus. Art.
B	28	1	0,195s	857	33	4	3	2
D	23	1	0,119s	590	25	3	3	3
E	21	1	0,103s	496	22	3	2	2

Verbesserungen für $t = 0,6$

Autor	g-Index	max. Verb.	Laufzeit	$\|E\|$	$\|\mathcal{P}\|$	M_{max}	#Extr.	#versch. zus. Art.
B	28	1	0,163s	857	36	3	1	1
E	21	1	0,122s	496	22	3	2	2

Verbesserungen für $t = 0,7$

Autor	g-Index	max. Verb.	Laufzeit	$\|E\|$	$\|\mathcal{P}\|$	M_{max}	#Extr.	#versch. zus. Art.
E	21	1	0,127s	496	22	3	2	2

Die Ergebnisse für die Partitionierung der atomaren Artikel nach Auswahl der kleinsten Cliquen zuerst sind folgende:

Verbesserungen für $t = 0,4$

Autor	g-Index	max. Verb.	Laufzeit	$\|E\|$	$\|\mathcal{P}\|$	M_{max}	#Extr.	#versch. zus. Art.
B	28	1	0,229s	857	33	3	2	2
D	23	1	0,138s	590	25	3	3	3

Verbesserungen für $t = 0,5$

Autor	g-Index	max. Verb.	Laufzeit	$\|E\|$	$\|\mathcal{P}\|$	M_{max}	#Extr.	#versch. zus. Art.
B	28	1	0,211s	857	34	2	1	1
D	23	1	0,106s	590	25	3	5	5

Verbesserungen für $t = 0,6$

Autor	g-Index	max. Verb.	Laufzeit	$\|E\|$	$\|\mathcal{P}\|$	M_{max}	#Extr.	#versch. zus. Art.
B	28	1	$0,198$s	857	36	3	2	2

Verbesserungen für $t = 0,7$

Autor	g-Index	max. Verb.	Laufzeit	$\|E\|$	$\|\mathcal{P}\|$	M_{max}	#Extr.	#versch. zus. Art.
E	21	1	$0,113$s	496	23	3	2	2

7 Fazit und Ausblick

Es stellte sich zwar heraus, dass die g-Index Manipulation durch Artikel-Extraktionen für das von Google Scholar verwendete Unioncite NP-schwer ist, allerdings konnte wir durch unsere Experimente aufzeigen, dass die benötigte Rechenleistung in der Praxis äußerst klein ist. Der exponentielle Teil der Laufzeit, des hierfür entwickelten Algorithmus, wird nämlich durch das Minimum der Größe des größten zusammengefügten Artikels M_{max} und dem zu erreichenden g-Index bestimmt. Die zusammengefügten Artikel sind in realen Anwendungsfällen jedoch begrenzt großen Ausmaßes. Wenn wir demnach also annehmen würden der Wert von M_{max} wäre konstant, erhielte man für beliebige Größen dieser Konstante die selbe polynomielle Laufzeit. Zwar wäre die Laufzeit dann immer noch größer als die lineare Laufzeit bei der h-Index-Manipulation durch Artikel-Extraktionen, für die Praxis wäre sie jedoch noch praktikabel.

Ebenfalls scheint eine g-Index-Manipulation für Unioncite im Vergleich zu seinem h-Index Pendant jedoch relativ situational zu sein und wenn man so will resistenter dagegen. In Anbetracht dessen, dass sie davon abhängt, dass einzelne zitierende Artikel mehrere atomare Artikel im selben zusammengefügten Artikel zitieren, sollte man in Frage stellen wie üblich es ist, dass einzelne Artikel in der Realität mehrere Versionen des selben Artikels zitieren. Vorstellbar wäre dies lediglich, wenn explizit die verschiedenen Versionen referenziert werden sollen, oder das Zusammenfügen in einer unbeabsichtigten Art und Weise verwendet wird. Wenn also durch den Algorithmus besonders große Verbesserungen möglich sind, kann dies als Indiz für eine vorherige Manipulation des Profils erachtet werden. Dementsprechend wäre es interessant die g-Index-Manipulation durch Extraktion im Zusammenspiel mit einer vorherigen g-Index- oder h-Index-manipulierenden Zusammenfüge-Operation zu betrachten. So könnte ein Autor im Laufe der Zeit zwischen verschiedenen Methoden alternieren, um aus neu hinzugekommen Zitationen zu profitieren. Generell sollte der Einfluss der Extraktionen auf den h-Index in Relation zum g-Index untersucht werden.

Literatur

[1] AIs 10 to watch. *IEEE Intelligent Systems*, 26(1):5–15, jan 2011. 36

[2] Google scholar citations open to all. `https://scholar.googleblog.com/2011/11/google-scholar-citations-open-to-all.html`, Nov 2011. 4

[3] René Van Bevern, Christian Komusiewicz, Rolf Niedermeier, Manuel Sorge, and Toby Walsh. H-index manipulation by merging articles: Models, theory, and experiments. *Artificial Intelligence*, 240:19–35, 2016. 5, 6, 7, 36, 37

[4] René Van Bevern, Christian Komusiewicz, Rolf Niedermeier, Manuel Sorge, and Toby Walsh. h-index manipulation by undoing merges. *Frontiers in Artificial Intelligence and Applications*, 285(ECAI 2016):895–903, 2016. 5, 7, 9

[5] Bart de Keijzer and Krzysztof R. Apt. The h-index can be easily manipulated. *Bulletin of the EATCS*, 110:79–85, 2013. 5

[6] Leo Egghe. Theory and practise of the g-index. *Scientometrics*, 69(1):131–152, 2006. 4, 8

[7] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002. 18

[8] Jorge 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*, pages 16569–16572, 2005. 4, 8

[9] Chrystalla Pavlou and Edith Elkind. Manipulating citation indices in a social context. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 32–40. International Foundation for Autonomous Agents and Multiagent Systems, 2016. 5

[10] Daniel Zeng. AIs 10 to watch. *IEEE Intelligent Systems*, 28(3):86–96, may 2013. 36