Technische Universität Berlin
Electrical Engineering and Computer Science
Institute of Software Engineering and Theoretical Computer Science
Algorithmics and Computational Complexity (AKT)

A Simple and Robust Measure of Triadic Closure: Algorithmic and Structural Aspects

Sebastian Bruchhold (410586)

Thesis submitted in fulfillment of the requirements for the degree
"Bachelor of Science" (B. Sc.) in the field of Computer Science

September 2022

Supervisor and first reviewer:	Dr. André Nichterlein
Second reviewer:	Prof. Dr. Stefan Schmid
Co-Supervisor:	Tomohiro Koana

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.

Die selbstständige und eigenhändige Ausfertigung versichert an Eides statt

Berlin, den ____________________ ____________________

Datum Unterschrift

Zusammenfassung

Das *c-Closure* eines Graphen wurde von Fox et al. [SICOMP '20] eingeführt und misst *Triadic Closure*: Dies ist die Tendenz zweier Knoten, adjazent zu sein, wenn sie viele gemeinsame Nachbarn haben. Für einen Graphen $G = (V, E)$ kann das c-Closure mittels *Closure-Numbers* definiert werden, welche durch Koana et al. [ISAAC '20] eingeführt wurden. Wir passen deren Definition leicht an. Die Closure-Number eines Knotens $v \in V$ ist wie folgt definiert:

$$\mathrm{cl}_G(v) \coloneqq \max_{u \in V \setminus N_G[v]} \{|N_G(u) \cap N_G(v)|\} + 1$$

Wenn $V \setminus N_G[v] = \varnothing$, dann $\mathrm{cl}_G(v) \coloneqq 1$. So lässt sich das c-Closure von G als die größte Closure-Number definieren:

$$\mathrm{c}(G) \coloneqq \max_{v \in V} \{\mathrm{cl}_G(v)\}$$

Das c-Closure hat mehrere wünschenswerte Qualitäten, darunter seine Einfachheit. Darüber hinaus wurde es bereits erfolgreich zum Entwurf parametrisierter Algorithmen verwendet. Jedoch ist das c-Closure nicht sehr robust, da das Entfernen einer Kante zu einem unbeschränkten Anstieg im c-Closure führen kann.

Es ist das Ziel dieser Arbeit, ein Maß für Triadic Closure einzuführen, das robuster ist als das c-Closure, aber dessen Einfachheit beibehält. Zu diesem Zweck benutzen wir den *h-Index*, welcher von Hirsch [PNAS '05] eingeführt wurde. Der h-Index einer nicht leeren, aber endlichen Multimenge $M \subseteq \mathbb{N}_0$ ist die eindeutige Zahl $\mathrm{h}(M) \in \mathbb{N}_0$, so dass es $\mathrm{h}(M)$ Werte in M gibt, die mindestens $\mathrm{h}(M)$ sind, während alle anderen Werte in M höchstens $\mathrm{h}(M)$ sind. Dann definieren wir den *hc-Index* von G als den h-Index von allen Closure-Numbers:

$$\mathrm{hc}(G) \coloneqq \mathrm{h}([\mathrm{cl}_G(v) \mid v \in V])$$

Auch führen wir den *Weak hc-Index* von G ein: Er ist die kleinste Zahl $\mathrm{whc}(G) \in \mathbb{N}_1$, so dass eine Menge $U \subsetneq V$ mit $|U| \leq \mathrm{whc}(G)$ und $\mathrm{c}(G[V \setminus U]) \leq \mathrm{whc}(G)$ existiert. Der Weak hc-Index ist nie größer als der hc-Index, und der hc-Index ist nie größer als das c-Closure.

In der Arbeit charakterisieren wir zunächst die Graphen mit kleinem hc-Index durch verbotene induzierte Teilgraphen, und wir ermitteln die Position des hc-Index und des Weak hc-Index in der Hierarchie der Graphenparameter. Danach zeigen wir, dass der hc-Index in Polynomialzeit berechnet werden kann und dass er in einigen Graphen aus der echten Welt relativ klein ist. Jedoch kann der Weak hc-Index nicht in Polynomialzeit berechnet werden, es sei denn, $P = NP$. Außerdem zeigen wir, dass für jede gegebene Zahl $k \in \mathbb{N}_1$ in

$$\mathcal{O}^*\big((\mathrm{hc}(G) \cdot k)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(\mathrm{hc}(G))}\big)$$

Zeit entschieden werden kann, ob G eine dominierende Menge der Größe höchstens k enthält, indem wir einen Algorithmus von Koana et al. [ESA '20] anpassen. Abschließend beweisen wir, dass für jede beliebige Konstante $k \in \mathbb{N}_1$ gilt: Wenn der Graph G einen Weak hc-Index von höchstens k hat, dann enthält er

$$\mathcal{O}\big(n_G^{2-2^{1-k}}\big)$$

maximale Cliquen. Somit passen wir eine Schranke von Fox et al. [SICOMP '20] an. Dies legt nahe, dass der (Weak) hc-Index generell nützlich ist, um Ergebnisse anzupassen, die vom c-Closure abhängen.

Abstract

The *c-closure* of a graph was introduced by Fox et al. [SICOMP '20], and it measures *triadic closure*: the tendency of two vertices to be adjacent if they have many common neighbors. For a graph $G = (V, E)$, its c-closure can be defined using *closure numbers*, which were introduced by Koana et al. [ISAAC '20]. We modify their definition slightly. The closure number of a vertex $v \in V$ is

$$\mathrm{cl}_G(v) \coloneqq \max_{u \in V \setminus N_G[v]} \{|N_G(u) \cap N_G(v)|\} + 1$$

if $V \setminus N_G[v] \neq \varnothing$; otherwise, $\mathrm{cl}_G(v) \coloneqq 1$. We may then define the c-closure of G as the maximum closure number:

$$\mathrm{c}(G) \coloneqq \max_{v \in V} \{\mathrm{cl}_G(v)\}$$

The c-closure has several desirable qualities, including its simplicity. Furthermore, it has been used successfully to design parameterized algorithms. However, the c-closure is not very robust, as removing an edge from a graph can lead to an unbounded increase in c-closure.

It is the intent of this thesis to introduce a measure of triadic closure that is more robust than the c-closure while maintaining its simplicity. For this purpose, we use the *h-index*, which was introduced by Hirsch [PNAS '05]. The h-index of a non-empty and finite multiset $M \subseteq \mathbb{N}_0$ is the unique number $\mathrm{h}(M) \in \mathbb{N}_0$ such that there are $\mathrm{h}(M)$ values in M that are at least $\mathrm{h}(M)$, while all other values in M are at most $\mathrm{h}(M)$. Then, we define the *hc-index* of G to be the h-index of all closure numbers:

$$\mathrm{hc}(G) \coloneqq \mathrm{h}([\mathrm{cl}_G(v) \mid v \in V])$$

We also introduce the *weak hc-index* of G: It is the smallest number $\mathrm{whc}(G) \in \mathbb{N}_1$ such that there exists a set $U \subsetneq V$ with $|U| \leq \mathrm{whc}(G)$ and $\mathrm{c}(G[V \setminus U]) \leq \mathrm{whc}(G)$. The weak hc-index is never greater than the hc-index, and the hc-index is never greater than the c-closure.

In the thesis, we first characterize the graphs of small hc-index in terms of forbidden induced subgraphs, and we establish the position of the hc-index and the weak hc-index in the graph parameter hierarchy. Then, we show that the hc-index can be computed in polynomial time and that it is relatively small in some real-world graphs. However, the weak hc-index cannot be computed in polynomial time unless $P = NP$. We further show that given a number $k \in \mathbb{N}_1$, deciding if G contains a dominating set of size at most k takes

$$\mathcal{O}^*\big((\mathrm{hc}(G) \cdot k)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(\mathrm{hc}(G))}\big)$$

time, adapting an algorithm by Koana et al. [ESA '20]. Finally, we prove that for any fixed $k \in \mathbb{N}_1$, if G has a weak hc-index of at most k, then G contains

$$\mathcal{O}\big(n_G^{2-2^{1-k}}\big)$$

maximal cliques, adapting a bound by Fox et al. [SICOMP '20]. This suggests that the (weak) hc-index is generally useful to adapt results that rely on the c-closure.

Contents

Chapter 1

Introduction

This chapter provides an overview of the thesis, covering motivation and related work.

1.1 Motivation

In social networks, two individuals with many common friends tend to be friends themselves [Gra73]. This property, known as *triadic closure*, motivated Fox et al. [Fox+20] to introduce the *c-closure* of a graph $G = (V, E)$: It is the smallest number $\mathrm{c}(G) \in \mathbb{N}_1$ such that any two distinct vertices with at least $\mathrm{c}(G)$ common neighbors are adjacent. Based on this definition, it is to be expected that the c-closure is small in graphs that exhibit high triadic closure, that is, graphs where two vertices are likely to be adjacent if they share many neighbors.

Koana et al. [KKS20a] use an alternative definition of c-closure, which we modify slightly: First, we define the *closure number* of a vertex $v \in V$ as

$$\mathrm{cl}_G(v) \coloneqq \max_{u \in V \setminus N_G[v]} \{|N_G(u) \cap N_G(v)|\} + 1$$

if $V \setminus N_G[v]$ is non-empty; otherwise, we define that $\mathrm{cl}_G(v) \coloneqq 1$. Then, we can define the c-closure as the maximum closure number:

$$\mathrm{c}(G) \coloneqq \max_{v \in V} \{\mathrm{cl}_G(v)\}$$

Throughout the thesis, we will rely on these definitions. Especially the closure numbers play an important role since they let us identify vertices that violate the triadic closure property.

As a graph parameter, the c-closure has several desirable qualities; for example, it is simple to understand and compute [KKS20b]. The c-closure is also simple in the sense that for any $c \in \mathbb{N}_1$, it is easy to characterize the graphs of c-closure at most c in terms of forbidden induced subgraphs. The graphs of c-closure at most 2, for instance, are the graphs that contain neither an induced cycle on four vertices nor an induced diamond, as Fox et al. [Fox+20] note. A drawing of the diamond is shown in Figure 1.1. Additionally, the c-closure can be used to bound the number of maximal cliques in a graph [Fox+20]. Furthermore, it is useful in the design of parameterized algorithms, including algorithms for finding dominating sets [KKS20b].

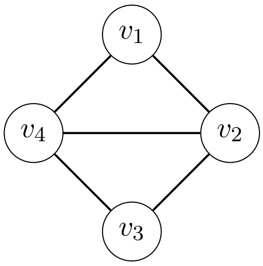

Figure 1.1: The *diamond*, denoted D.

However, the c-closure is not very robust, as removing an edge from a graph can lead to an unbounded increase in c-closure: Consider the complete graph K_n for any $n \in \mathbb{N}_1$ with $n \geq 3$. While we have $\mathrm{c}(K_n) = 1$, the c-closure increases to $\mathrm{c}(K_n - e) = n - 1$ after removing an edge because the resulting graph $K_n - e$ contains two non-adjacent vertices that share $n - 2$ neighbors.

As a more robust alternative, Fox et al. [Fox+20] introduce the *weak c-closure*, which can be defined using closure numbers [KKS20a]: It is the smallest number $\mathrm{wc}(G) \in \mathbb{N}_1$ such that for every non-empty set $U \subseteq V$, the induced subgraph $G[U]$ contains at least one vertex $v \in U$ with $\mathrm{cl}_{G[U]}(v) \leq \mathrm{wc}(G)$.

To give an example, we calculate the weak c-closure of the diamond D. Since the vertices v_2 and v_4 are each adjacent to all other vertices, we have $\mathrm{cl}_D(v_2) = \mathrm{cl}_D(v_4) = 1$. More generally, this applies to every induced subgraph of D that contains v_2 or v_4. It remains to consider the induced subgraphs that contain neither v_2 nor v_4. Such induced subgraphs will consist only of isolated vertices, because v_1 and v_3 are not adjacent. As the closure number of an isolated vertex is 1, it follows that all induced subgraphs of D contain a vertex of closure number 1, and therefore $\mathrm{wc}(D) = 1$. By a similar argument, we can deduce that $\mathrm{wc}(K_n - e) = 1$ for all $n \in \mathbb{N}_1$. In contrast, $\mathrm{c}(D) = \mathrm{c}(K_4 - e) = 3$ since $\mathrm{cl}_D(v_1) = \mathrm{cl}_D(v_3) = 3$.

Like the c-closure, the weak c-closure may be used to bound the number of maximal cliques in a graph [Fox+20]. For graphs of bounded weak c-closure, Fox et al. [Fox+20] prove a bound that is quadratic in the number of vertices, whereas for graphs of bounded c-closure, they prove a bound that is subquadratic. Contrast this with arbitrary graphs, which may contain exponentially many maximal cliques [MM65].

Similarly, the weak c-closure can also be used to design parameterized algorithms; for instance, finding a dominating set of at most a given solution size is fixed-parameter tractable when combining the solution size and the weak c-closure [LS21]. However, as for any given $c \in \mathbb{N}_1$, the class of graphs with a weak c-closure of at most c is generally much broader than the class of graphs with a c-closure of at most c, designing algorithms that exploit the weak c-closure is more difficult. Continuing the example, the c-closure admits a simpler algorithm for finding dominating sets [KKS20b].

Also, it appears to be difficult to characterize the graphs of small weak c-closure in terms of forbidden induced subgraphs. Koana et al. [KKS20a] note that a graph has a weak c-closure of 1 if and only if it contains neither an induced cycle on four vertices nor an induced path on four vertices. For the graphs of weak c-closure at most 2, however, they note that finding such a characterization remains an open problem.

This thesis attempts to introduce a measure of triadic closure that is more robust than the c-closure while maintaining its simplicity. To be specific, we want to introduce a graph parameter that bounds most of the closure numbers in a graph but tolerates a few vertices with high closure numbers. For this purpose, we can use the *h-index* of a non-empty, finite multiset $M \subseteq \mathbb{N}_0$, which was introduced by Hirsch [Hir05]: It is the unique number $\mathrm{h}(M) \in \mathbb{N}_0$ such that there are $\mathrm{h}(M)$ values in M that are at least $\mathrm{h}(M)$, while all other values in M are at most $\mathrm{h}(M)$. Then, we define the *hc-index* of G to be the h-index of all closure numbers:

$$\mathrm{hc}(G) := \mathrm{h}([\mathrm{cl}_G(v) \mid v \in V])$$

We use the diamond D as an example again. We have already established that the closure numbers are $\mathrm{cl}_D(v_1) = \mathrm{cl}_D(v_3) = 3$ and $\mathrm{cl}_D(v_2) = \mathrm{cl}_D(v_4) = 1$. Therefore, the hc-index of D is the h-index of the multiset $[3, 1, 3, 1]$, so $\mathrm{hc}(D) = 2$.

The hc-index is robust in the sense that removing an edge from a graph can increase its hc-index by at most 2: Let $G' = (V, E')$ be the graph that is obtained after removing an edge $\{u, v\} \in E \setminus E'$ from G. The closure number of any vertex from $V \setminus \{u, v\}$ will not increase. Because G contains at most $\mathrm{hc}(G)$ vertices with a closure number greater than $\mathrm{hc}(G)$, the graph G' contains at most $\mathrm{hc}(G) + 2$ vertices with closure numbers that are greater than $\mathrm{hc}(G)$. Hence, $\mathrm{hc}(G') \leq \mathrm{hc}(G) + 2$.

Further, we introduce the *weak hc-index*, which is the smallest number $\mathrm{whc}(G) \in \mathbb{N}_1$ such that there exists a set $U \subsetneq V$ with $|U| \leq \mathrm{whc}(G)$ and $\mathrm{c}(G[V \setminus U]) \leq \mathrm{whc}(G)$. For every $k \in \mathbb{N}_1$, the class of graphs with weak hc-index at most k fully contains the class of graphs with hc-index at most k, as the weak hc-index demands low closure numbers only after removing the vertices of high closure number.

In the diamond D, it suffices to remove the vertex v_1 to obtain a complete graph, which has c-closure 1. Therefore, we conclude that $\mathrm{whc}(D) = 1$. In summary, the weak c-closure and the weak hc-index of D are 1, its hc-index is 2, and its c-closure is 3. We discuss the relationships between these parameters more thoroughly for general graphs in Chapter 4.

The hc-index is the main focus of this thesis, and we will show that multiple of the desirable properties of the c-closure carry over to the hc-index. This includes the fact that the class of graphs with hc-index 1 as well as the class of graphs with hc-index at most 2 is easy to characterize in terms of forbidden induced subgraphs (Chapter 3). We also show that the hc-index is simple to compute and relatively small in selected graphs from the real world (Chapter 5). In addition, we show that the hc-index is useful for the design of parameterized algorithms: We adapt the algorithm for finding dominating sets by Koana et al. [KKS20b] so as to exploit the hc-index (Chapter 6).

For the weak hc-index, we show that it is probably not efficiently computable, since it cannot be computed in polynomial time unless $P = NP$ (Chapter 5). Still, the weak hc-index allows us to bound the number of maximal cliques: We show that in graphs of bounded weak hc-index, the maximum number of maximal cliques is subquadratic in the number of vertices (Chapter 7), adapting the bound by Fox et al. [Fox+20]. From this, it follows that the problem of listing all maximal cliques in a graph is fixed-parameter tractable with respect to the weak hc-index.

1.2 Related work

Koana et al. [KKS20b] show that given a graph G and a number $k \in \mathbb{N}_1$, deciding if G contains a dominating set of size at most k takes $\mathcal{O}^*((\mathrm{c}(G) \cdot k)^{\mathcal{O}(k)})$ time. In Chapter 6, we follow their approach closely to obtain an algorithm for finding dominating sets that exploits the hc-index.

Lokshtanov and Surianarayanan [LS21] present an algorithm that decides whether a graph G contains a dominating set of size at most k in $\mathcal{O}^*(k^{\mathcal{O}(\mathrm{wc}(G)^2 \cdot k^3)})$ time given any number $k \in \mathbb{N}_1$. Because the weak c-closure is never greater than the hc-index, their algorithm is also suitable for graphs of small hc-index. This algorithm, however, is more complicated than the algorithm by Koana et al. [KKS20b], which we adapt to obtain a simpler algorithm for graphs of small hc-index (Chapter 6).

Downey and Fellows [DF95] show that deciding if a graph contains a dominating set of size k is $W[2]$-complete when parameterized by the solution size $k \in \mathbb{N}_1$ alone. They therefore conjecture that the problem is not fixed-parameter tractable. Fixed-parameter algorithms are known when combining the solution size with other parameters, as the results mentioned above demonstrate.

Fox et al. [Fox+20] show that for all $n, c \in \mathbb{N}_1$, a graph on n vertices with a c-closure of at most c contains at most $4^{(c+4)\cdot(c-1)/2} \cdot n^{2-2^{1-c}}$ maximal cliques, and they show that a graph with n vertices and a weak c-closure of at most c contains at most $3^{(c-1)/3} \cdot n^2$ maximal cliques. The weak hc-index admits similar bounds (Chapter 7).

Eppstein et al. [ELS10] prove bounds on the number of maximal cliques in graphs of bounded *degeneracy*, which is a graph parameter introduced by Lick and White [LW70]: For an arbitrary graph $G = (V, E)$, it is defined as the smallest number $\mathrm{d}(G) \in \mathbb{N}_0$ such that for every non-empty set $U \subseteq V$, the induced subgraph $G[U]$ contains at least one vertex $v \in U$ with $\deg_{G[U]}(v) \leq \mathrm{d}(G)$. In graphs of bounded degeneracy, the maximum number of maximal cliques is linear in the number of vertices [ELS10]. As we show in Chapter 4, the weak hc-index and the degeneracy are incomparable.

Moon and Moser [MM65] show that for all $n \in \mathbb{N}_1$, there exists a graph on n vertices with $3^{\lfloor n/3 \rfloor}$ maximal cliques. Contrasting this with the bounds mentioned above, graphs of small degeneracy or small (weak) c-closure contain relatively few maximal cliques in comparison.

We recall that the hc-index is the h-index of all closure numbers. This definition is similar to a definition by Eppstein and Spiro [ES09], who define the *h-index* of a graph to be the h-index of all degrees. The introduction of this parameter was also motivated by the structure of social networks. In Chapter 4, we show that the hc-index of a graph is at most one greater than its h-index.

Chapter 2

Preliminaries

This chapter is used to define preliminary terms and notation. Unless indicated otherwise, most of the definitions for standard concepts from graph theory are based on the textbook by Diestel [Die00] or Bondy and Murty [BM08]. In addition, some familiarity with concepts from parameterized complexity and algorithmics is assumed [Nie06].

Basic graph notation. Let $G = (V, E)$ denote a (*simple*) *graph*, where V denotes the non-empty, finite set of *vertices* and $E \subseteq \binom{V}{2}$ denotes the set of *edges*.

Now, let $v \in V$ be an arbitrary vertex, and let $U \subseteq V$ be any subset of the vertices. Then, we denote by

$V(G)$	the *vertex set* of G, formally, $V(G) := V$;
$E(G)$	the *edge set* of G, formally, $E(G) := E$;
n_G	the *order* of G, formally, $n_G := \lvert V \rvert$;
m_G	the *size* of G, formally, $m_G := \lvert E \rvert$;
$N_G(v)$	the (*open*) *neighborhood* of v, formally, $N_G(v) := \{u \in V \mid \{u, v\} \in E\}$;
$N_G(U)$	the (*open*) *neighborhood* of U, formally, $N_G(U) := (\bigcup_{u \in U} N_G(u)) \setminus U$;
$N_G[v]$	the *closed neighborhood* of v, formally, $N_G[v] := N_G(v) \cup \{v\}$;
$N_G[U]$	the *closed neighborhood* of U, formally, $N_G[U] := N_G(U) \cup U$;
$\deg_G(v)$	the *degree* of v, formally, $\deg_G(v) := \lvert N_G(v) \rvert$;
$\Delta(G)$	the *maximum degree* of G, formally, $\Delta(G) := \max_{u \in V}\{\deg_G(u)\}$;
$G[U]$	the *induced subgraph* on $U \neq \varnothing$, formally, $G[U] := (U, E \cap \binom{U}{2})$;
$\overline{G}$	the *complement* of G, formally, $\overline{G} := (V, \binom{V}{2} \setminus E)$.

Basic graph terms. Let $G = (V, E)$ be a graph, and let $U \subseteq V$ be a non-empty set. If two vertices $u, v \in V$ are connected by an edge $\{u, v\} \in E$, then u and v are said to be *adjacent*. If every pair of distinct vertices in U is adjacent, then U is called a *clique*. If no pair of vertices in U is adjacent, then U is called an *independent set*. If the closed neighborhood of U contains all vertices of G, then U is called a *dominating set*. A set of vertices $W \subseteq V$ is called a *vertex cover* if $e \cap W \neq \varnothing$ for every edge $e \in E$.

Let H be a second graph. If H can be obtained by injectively relabeling the vertices of G, then the two graphs are said to be *isomorphic*, which we denote as $G \simeq H$. Their *union* is defined as $G \cup H \coloneqq (V \cup V(H), E \cup E(H))$, and the *disjoint union* is defined as $G \uplus H \coloneqq G \cup H'$, where H' is some graph with $H' \simeq H$ and $V \cap V(H') = \varnothing$. For a positive integer $n \in \mathbb{N}_1$, we denote the disjoint union of n copies of G as $n \cdot G$.

Named graphs. Let $m, n \in \mathbb{N}_1$ be positive integers. We denote by

K_n the *complete graph* of order n, formally, $K_n \coloneqq ([n], \binom{[n]}{2})$;

$K_n - e$ the *near-complete graph* of order n, formally, $K_n - e \coloneqq ([n], \binom{[n]}{2} \setminus \{\{1, 2\}\})$;

$K_{m,n}$ the *complete bipartite graph* of order $m + n$, formally, $K_{m,n} \coloneqq \overline{(K_m \uplus K_n)}$;

P_3 the 3-*path*, formally, $P_3 \coloneqq K_3 - e$;

C_4 the 4-*cycle*, formally, $C_4 \coloneqq K_{2,2}$;

D the *diamond*, formally, $D \coloneqq K_4 - e$.

Graph parameters. Koana et al. [KKS20a] define the *closure number* of a vertex in a graph. In this thesis, a slightly modified definition is used.

Definition 2.1. The *closure number* of a vertex $v \in V$ in a graph $G = (V, E)$ is:

$$\mathrm{cl}_G(v) \coloneqq \max_{u \in V \setminus N_G[v]} \{|N_G(u) \cap N_G(v)|\} + 1$$

If there is no vertex in V that is distinct from and non-adjacent to v, then $\mathrm{cl}_G(v) \coloneqq 1$.

Fox et al. [Fox+20] introduce the *c-closure* and the *weak c-closure* of a graph. They can be defined using closure numbers [KKS20a].

Definition 2.2. The *c-closure* of $G = (V, E)$ is the maximum closure number:

$$\mathrm{c}(G) \coloneqq \max_{v \in V} \{\mathrm{cl}_G(v)\}$$

Definition 2.3. The *weak c-closure* of $G = (V, E)$ is the smallest number $\mathrm{wc}(G) \in \mathbb{N}_1$ such that for every non-empty set $U \subseteq V$, the induced subgraph $G[U]$ contains at least one vertex $v \in U$ with $\mathrm{cl}_{G[U]}(v) \leq \mathrm{wc}(G)$.

The definition above is similar to the definition of the *degeneracy* of a graph, which was introduced by Lick and White [LW70].

Definition 2.4. The *degeneracy* of $G = (V, E)$ is the smallest number $\mathrm{d}(G) \in \mathbb{N}_0$ such that for every non-empty set $U \subseteq V$, the induced subgraph $G[U]$ contains at least one vertex $v \in U$ with $\deg_{G[U]}(v) \leq \mathrm{d}(G)$.

Hirsch [Hir05] introduces the index on which the next definition is based.

Definition 2.5. The *h-index* of a non-empty and finite multiset $M \subseteq \mathbb{N}_0$ is the unique number $\mathrm{h}(M) \in \mathbb{N}_0$ such that there are $\mathrm{h}(M)$ values in M that are at least $\mathrm{h}(M)$, while all other values in M are at most $\mathrm{h}(M)$.

Eppstein and Spiro [ES09] apply this index to graphs using the definition below.

Definition 2.6. The *h-index* of a graph $G = (V, E)$ is the h-index of all degrees:

$$\mathrm{h}(G) \coloneqq \mathrm{h}([\deg_G(v) \mid v \in V])$$

Combining h-index and c-closure, the following parameters are introduced here.

Definition 2.7. The *hc-index* of $G = (V, E)$ is the h-index of all closure numbers:

$$\mathrm{hc}(G) \coloneqq \mathrm{h}([\mathrm{cl}_G(v) \mid v \in V])$$

Definition 2.8. The *weak hc-index* of $G = (V, E)$ is the smallest number $\mathrm{whc}(G) \in \mathbb{N}_1$ such that there exists a set $U \subsetneq V$ with $|U| \leq \mathrm{whc}(G)$ and $\mathrm{c}(G[V \setminus U]) \leq \mathrm{whc}(G)$.

Chapter 3

Graphs of small hc-index

For any $c \in \mathbb{N}_1$, it is easy to see how the class of graphs with c-closure at most c can be characterized in terms of forbidden induced subgraphs. To give an example, the graphs with c-closure at most 2 are exactly the (C_4, D)-free graphs [Fox+20], where C_4 denotes the 4-cycle and D denotes the diamond. This class in particular has been studied for its own sake [Esc+11].

Characterizing the graphs of small weak c-closure appears to be less simple: While a forbidden induced subgraph characterization for the graphs with a weak c-closure of 1 is known, characterizing the graphs of weak c-closure at most 2 remains an open problem, as Koana et al. [KKS20a] note.

In this chapter, we characterize the graphs of small hc-index, in particular, hc-index exactly 1 (Theorem 3.2) and hc-index at most 2 (Theorem 3.3 and Figure 3.1). For this purpose, the following lemma is useful.

Lemma 3.1. *For all $k \in \mathbb{N}_1$, the class of graphs with an hc-index of at most k is closed under taking induced subgraphs.*

Proof. Let $k \in \mathbb{N}_1$ be a positive integer, let $G = (V, E)$ be a graph with an hc-index of at most k, and let $U \subseteq V$ be a non-empty set of vertices. For the sake of contradiction, assume that the induced subgraph $G[U]$ has an hc-index that is greater than k. Hence, there are more than k vertices with closure numbers greater than k in $G[U]$. Since any two distinct, non-adjacent vertices in $G[U]$ are non-adjacent in G and share at least as many neighbors in G as they do in $G[U]$, there are more than k vertices with a closure number that is greater than k in G. This is a contradiction to G having an hc-index of at most k. Therefore, the hc-index of $G[U]$ is at most k as well. □

3.1 Graphs of hc-index 1

First, we characterize the graphs of hc-index 1. We recall that P_3 denotes the 3-path.

Theorem 3.2. *A graph has hc-index 1 if and only if it does not contain an induced P_3.*

Proof. We prove the two directions individually.

(⇒) Let $G = (V, E)$ be a graph of hc-index 1. Assume, for the sake of contradiction, that there is a set of vertices $U \subseteq V$ such that $G[U] \simeq P_3$. Since $G[U]$ has hc-index 2 but G has hc-index 1, this contradicts Lemma 3.1. We infer that no induced subgraph isomorphic to P_3 exists in G.

($\Leftarrow$) We use contraposition to prove this direction. Let $G = (V, E)$ be a graph with an hc-index of at least 2. Then, there exists a vertex $v \in V$ with a closure number that is at least 2. Thus, there is another vertex $u \in V$ that is non-adjacent to v, and they have at least one common neighbor $w \in V$. Choose $U \coloneqq \{u, v, w\}$. It follows that $G[U] \simeq P_3$ because $E(G[U]) = \{\{v, w\}, \{w, u\}\}$. □

3.2 Graphs of hc-index at most 2

For the graphs of hc-index at most 2, we prove two characterizations and give a list of forbidden induced subgraphs, which was found by computer-assisted search.

Theorem 3.3. *A graph has hc-index at most* 2 *if and only if it contains no induced* C_4 *and at most one pair of distinct, non-adjacent vertices that is part of an induced* D.

Proof. The proof has two directions.

($\Rightarrow$) Let $G = (V, E)$ be a graph of hc-index at most 2. Because the hc-index of C_4 is 3, the graph G contains no induced subgraph isomorphic to C_4, as implied by Lemma 3.1. For the sake of contradiction, suppose that there are two distinct pairs of non-adjacent vertices $U_1, U_2 \in E(\overline{G})$ such that there are two distinct sets $W_1, W_2 \subseteq V$ with $U_1 \subseteq W_1$ and $U_2 \subseteq W_2$ satisfying $G[W_1] \simeq G[W_2] \simeq D$. Then, $\mathrm{cl}_G(v) \geq 3$ for all $v \in U_1 \cup U_2$. As the pairs U_1 and U_2 are distinct, there are at least three vertices in G that have closure numbers of at least 3. This is a contradiction to G having hc-index at most 2, and thus there is at most one pair of distinct, non-adjacent vertices that is part of an induced D.

($\Leftarrow$) We prove this direction using contraposition. Let $G = (V, E)$ be a graph with an hc-index of at least 3. By definition of the hc-index, the graph G contains at least three distinct vertices $v_1, v_2, v_3 \in V$ such that for every $i \in \{1, 2, 3\}$, we have $\mathrm{cl}_G(v_i) \geq 3$, and thus there exists a vertex $x_i \in V$ that is distinct from and non-adjacent to v_i, and they share at least two distinct neighbors $u_i, w_i \in N_G(v_i) \cap N_G(x_i)$. If for some $i \in \{1, 2, 3\}$, the common neighbors u_i and w_i are non-adjacent, then $G[\{u_i, v_i, w_i, x_i\}] \simeq C_4$ and we are done. Otherwise, $G[\{u_i, v_i, w_i, x_i\}] \simeq D$ for every $i \in \{1, 2, 3\}$. Because v_1, v_2, v_3 are distinct, at least two of the pairs $\{v_1, x_1\}, \{v_2, x_2\}, \{v_3, x_3\}$ are distinct. □

We recall that $G \cup H \coloneqq (V(G) \cup V(H), E(G) \cup E(H))$ for graphs G and H.

Corollary 3.4. *A graph has hc-index at most* 2 *if and only if it contains no induced* C_4 *and either no induced* D*, or the union over every induced* D *is isomorphic to* $K_n - e$ *for some* $n \in \mathbb{N}_1$.

Proof. We prove each direction separately.

($\Rightarrow$) Let $G = (V, E)$ be any graph of hc-index at most 2. Then, G does not contain an induced C_4. If G contains no induced D, then we are done. Otherwise, we consider the union over every induced D:

$$H \coloneqq \bigcup_{\substack{U \subseteq V \\ G[U] \simeq D}} G[U]$$

By Theorem 3.3, there is a pair of distinct, non-adjacent vertices $P \in E(\overline{G})$ such that for any two sets $W_1, W_2 \subseteq V$ with $G[W_1] \simeq G[W_2] \simeq D$, we have $P \subseteq W_1$ and $P \subseteq W_2$. Let $u \in W_1 \setminus P$ and $v \in W_2 \setminus P$ be distinct vertices. Then, u is adjacent to both of the vertices in P, and v is adjacent to both vertices in P. Consequently, u and v must be adjacent in G: Otherwise, we would have $G[P \cup \{u, v\}] \simeq C_4$, which is impossible. But instead, we have $G[P \cup \{u, v\}] \simeq D$, so u and v are also adjacent in H. It follows that any two distinct vertices from $V(H) \setminus P$ are adjacent in H. Therefore, $H \simeq K_n - e$ for some $n \in \mathbb{N}_1$.

($\Leftarrow$) Let G be any graph that contains no induced C_4 and either no induced D, or the union over every induced D is isomorphic to $K_n - e$ for some $n \in \mathbb{N}_1$. Clearly, there is at most one pair of distinct, non-adjacent vertices that is part of an induced D. Thus, the graph G has hc-index at most 2 by Theorem 3.3. □

Remark. Stating the inverse of Theorem 3.3, a graph has hc-index at least 3 if and only if it contains an induced C_4 or at least two pairs of distinct, non-adjacent vertices such that each pair is part of some induced D. Two such pairs, together with one induced D for each pair, will take up at most eight vertices. Therefore, the graphs with hc-index at most 2 can be characterized in terms of forbidden induced subgraphs with at most eight vertices. An exhaustive computer-assisted search was conducted, which has shown that any graph has hc-index at most 2 if and only if it does not contain an induced subgraph isomorphic to one of the following fifteen graphs:

$([4], \{\{3,1\},\{3,2\},\{4,1\},\{4,2\}\})$
$([5], \{\{2,1\},\{3,1\},\{3,2\},\{4,1\},\{4,2\},\{5,1\},\{5,2\}\})$
$([5], \{\{2,1\},\{3,1\},\{3,2\},\{4,1\},\{4,3\},\{5,1\},\{5,2\}\})$
$([5], \{\{2,1\},\{3,1\},\{3,2\},\{4,1\},\{4,2\},\{4,3\},\{5,1\},\{5,2\}\})$
$([7], \{\{2,1\},\{3,1\},\{4,1\},\{4,3\},\{5,1\},\{5,3\},\{6,1\},\{6,2\},\{7,1\},\{7,2\}\})$
$([7], \{\{2,1\},\{3,1\},\{4,1\},\{4,3\},\{5,3\},\{5,4\},\{6,1\},\{6,2\},\{7,1\},\{7,2\}\})$
$([7], \{\{2,1\},\{4,3\},\{5,1\},\{5,2\},\{5,3\},\{5,4\},\{6,3\},\{6,4\},\{7,1\},\{7,2\}\})$
$([7], \{\{3,1\},\{3,2\},\{4,2\},\{5,1\},\{5,4\},\{6,1\},\{6,4\},\{6,5\},\{7,1\},\{7,2\},\{7,3\}\})$
$([8], \{\{2,1\},\{4,3\},\{5,3\},\{5,4\},\{6,3\},\{6,4\},\{7,1\},\{7,2\},\{8,1\},\{8,2\}\})$
$([8], \{\{2,1\},\{3,1\},\{4,3\},\{5,3\},\{5,4\},\{6,3\},\{6,4\},\{7,1\},\{7,2\},\{8,1\},\{8,2\}\})$
$([8], \{\{2,1\},\{4,3\},\{5,1\},\{5,3\},\{5,4\},\{6,3\},\{6,4\},\{7,1\},\{7,2\},\{8,1\},\{8,2\}\})$
$([8], \{\{2,1\},\{4,3\},\{5,2\},\{5,3\},\{5,4\},\{6,1\},\{6,3\},\{6,4\},\{7,1\},\{7,2\},\{8,1\},\{8,2\}\})$
$([8], \{\{2,1\},\{4,3\},\{5,3\},\{5,4\},\{6,4\},\{6,5\},\{7,1\},\{7,2\},\{7,3\},\{8,1\},\{8,2\}\})$
$([8], \{\{2,1\},\{4,3\},\{5,3\},\{5,4\},\{6,1\},\{6,4\},\{6,5\},\{7,1\},\{7,2\},\{7,3\},\{8,1\},\{8,2\}\})$
$([8], \{\{2,1\},\{5,3\},\{5,4\},\{6,3\},\{6,4\},\{6,5\},\{7,1\},\{7,2\},\{7,4\},\{8,1\},\{8,2\},\{8,3\}\})$

Drawings of the graphs are shown in Figure 3.1. The order of the drawings was chosen to highlight similarities between the graphs and differs from the order in the list above, which instead gives the graphs in the order of their discovery by the computer.

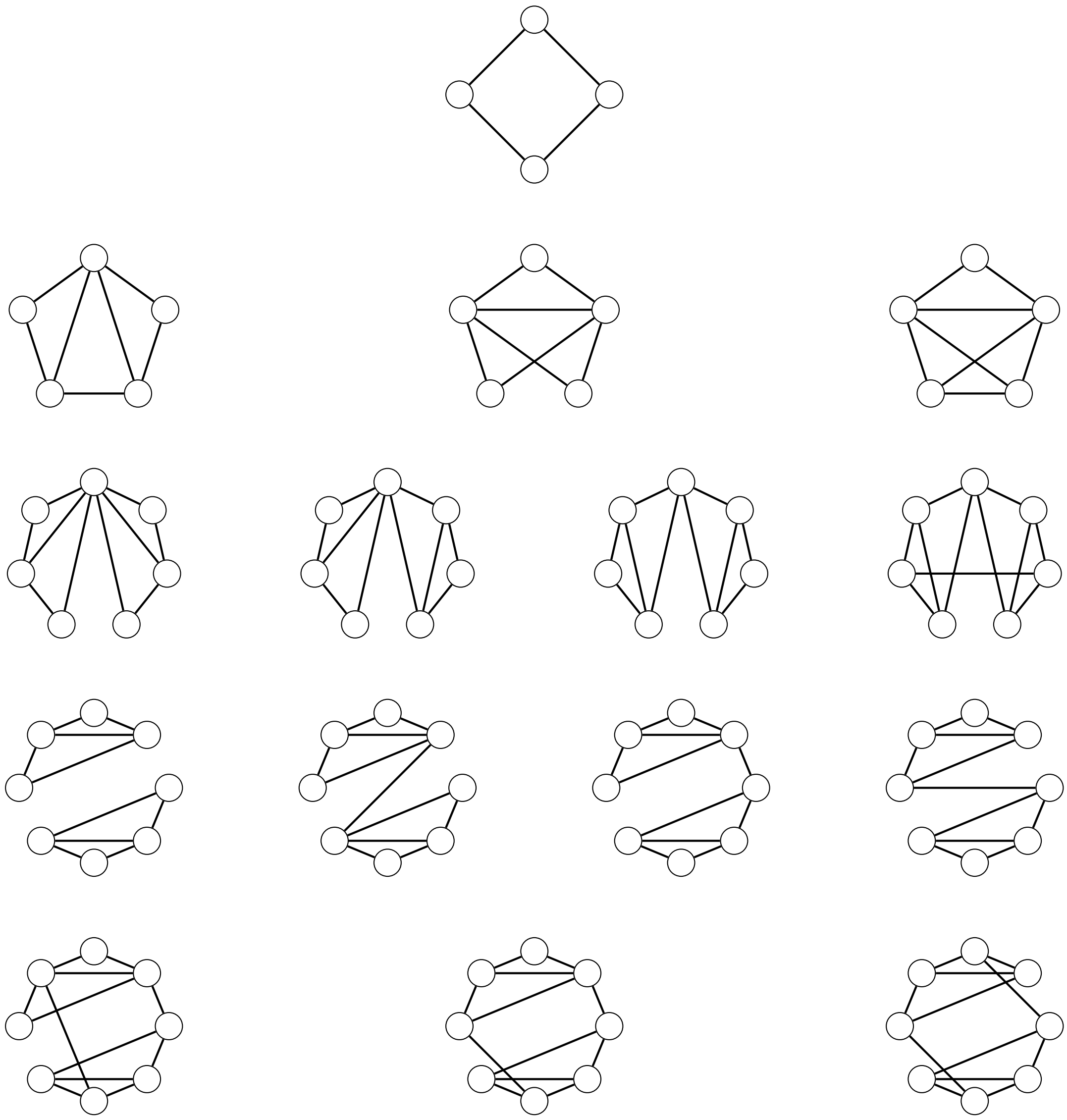

Figure 3.1: Forbidden induced subgraphs, found by computer-assisted search.

Chapter 4

Bounds on the hc-index

This chapter discusses the position of the hc-index and the weak hc-index in the graph parameter hierarchy. Concretely, we compare the two parameters with the h-index, the c-closure, the weak c-closure, and the degeneracy.

The relationships between those parameters are known: The weak c-closure is never greater than the c-closure [Fox+20]. It is also at most one greater than the degeneracy of a graph [KKS20a], but the c-closure and the degeneracy are incomparable [KKS20b]. Similarly, the c-closure and the h-index are incomparable [KN21]. The degeneracy is at most as great as the h-index, which is at most as great as the maximum degree [SW19]. Also, the c-closure is at most one greater than the maximum degree [KN21]. Several of these relationships are shown in a Hasse diagram by Koana et al. [Koa+22].

The results of this chapter expand this Hasse diagram (Figure 4.1). A line segment represents a bound on the lower parameter by a linear function of the upper parameter. Now, we prove the new relationships shown in Figure 4.1 one after the other. First, we prove the upper bounds on the hc-index. Then, we prove the lower bounds. Finally, we show the incomparability of multiple parameters.

Lemma 4.1. *The closure number of a vertex is at most one greater than its degree.*

Proof. Let $v \in V$ be a vertex of the graph $G = (V, E)$. Because $\deg_G(v) \geq 0$, the bound holds if $\mathrm{cl}_G(v) = 1$. Otherwise, if $\mathrm{cl}_G(v) \geq 2$, then $V \setminus N_G[v] \neq \varnothing$ and we can make the following deduction:

$$\mathrm{cl}_G(v) = \max_{u \in V \setminus N_G[v]} \{|N_G(u) \cap N_G(v)|\} + 1 \leq |N_G(v)| + 1 = \deg_G(v) + 1$$

Thus, $\mathrm{cl}_G(v) \leq \deg_G(v) + 1$. □

Theorem 4.2. *The hc-index of a graph is at most one greater than its h-index.*

Proof. Let $G = (V, E)$ be a graph. By definition of the h-index, there are $\mathrm{h}(G)$ vertices in V with degrees greater than or equal to $\mathrm{h}(G)$, and all other vertices in V have degrees less than or equal to $\mathrm{h}(G)$. Hence, there exists a set $U \subseteq V$ with $|U| = \mathrm{h}(G)$ such that for all $v \in W$, we have $\deg_G(v) \leq \mathrm{h}(G)$, where $W := V \setminus U$. By Lemma 4.1, it follows that $\mathrm{cl}_G(v) \leq \deg_G(v) + 1 \leq \mathrm{h}(G) + 1$ for all $v \in W$. Because $|W| = n_G - \mathrm{h}(G)$, there cannot be more than $\mathrm{h}(G) + 1$ vertices with closure numbers greater than $\mathrm{h}(G) + 1$, and so $\mathrm{hc}(G) \leq \mathrm{h}(G) + 1$. □

Theorem 4.3. *The hc-index of a graph is less than or equal to its c-closure.*

Proof. Let $G = (V, E)$ be any graph. By definition of the hc-index, there are $\mathrm{hc}(G) \geq 1$ vertices $v \in V$ with $\mathrm{cl}_G(v) \geq \mathrm{hc}(G)$. As $\mathrm{cl}_G(v) \leq \mathrm{c}(G)$, it follows that $\mathrm{hc}(G) \leq \mathrm{c}(G)$. $\square$

Theorem 4.4. *The weak c-closure of a graph is less than or equal to its hc-index.*

Proof. Let $G = (V, E)$ be a graph, and let $U \subseteq V$ be any non-empty set of vertices. We need to show that there exists some vertex $v \in U$ with $\mathrm{cl}_{G[U]}(v) \leq \mathrm{hc}(G)$. If there is a vertex $v \in U$ with $\mathrm{cl}_G(v) \leq \mathrm{hc}(G)$, then $\mathrm{cl}_{G[U]}(v) \leq \mathrm{hc}(G)$ and we are done. Otherwise, we know that $\mathrm{cl}_G(v) > \mathrm{hc}(G)$ for all $v \in U$. By definition of the hc-index, there are at most $\mathrm{hc}(G)$ such vertices v in G, and therefore $|U| \leq \mathrm{hc}(G)$. Hence, every vertex $v \in U$ has $\mathrm{cl}_{G[U]}(v) \leq \mathrm{c}(G[U]) \leq |U| \leq \mathrm{hc}(G)$. We infer that $\mathrm{wc}(G) \leq \mathrm{hc}(G)$. $\square$

Theorem 4.5. *The weak hc-index of a graph is less than or equal to its hc-index.*

Proof. Let $G = (V, E)$ be any graph. Then, choose $U := \{v \in V \mid \mathrm{cl}_G(v) > \mathrm{hc}(G)\}$. By definition of the hc-index, we have $|U| \leq \mathrm{hc}(G)$. Clearly, $\mathrm{c}(G[V \setminus U]) \leq \mathrm{hc}(G)$, and we deduce that $\mathrm{whc}(G) \leq \mathrm{hc}(G)$. $\square$

Theorem 4.6. *The hc-index and the degeneracy are incomparable.*

Proof. Let $n \in \mathbb{N}_1$ be any positive integer. Choose $G_n := n \cdot K_{2,n}$. The hc-index of G_n is greater than n because there are at least $2 \cdot n$ vertices with closure numbers greater than n, but the degeneracy of G_n is at most 2. In contrast, the hc-index of K_n is 1 by Theorem 3.2, but the degeneracy of K_n is $n - 1$. We thus conclude that neither of the two parameters bounds the other. $\square$

Theorem 4.7. *The weak hc-index and the degeneracy are incomparable.*

Proof. Let $n \in \mathbb{N}_1$ be any positive integer. Again, choose $G_n := n \cdot K_{2,n}$. By removing fewer than n vertices from G_n, the graph that is obtained has c-closure greater than n; hence, the weak hc-index of G_n is at least n, whereas the degeneracy is at most 2. The complete graph K_n, however, has weak hc-index 1 and degeneracy $n - 1$. $\square$

Theorem 4.8. *The weak hc-index and the weak c-closure are incomparable.*

Proof. Let $n \in \mathbb{N}_1$ be a positive integer. Choose $G_n := \overline{(K_{1,n} \uplus K_{1,n})}$. It is sufficient to remove two vertices from G_n to obtain a complete graph, which has c-closure 1. Thus, the weak hc-index of G_n is at most 2. However, because every closure number in G_n is greater than n, the weak c-closure is greater than n as well. Furthermore, $n \cdot K_{2,n}$ has a weak hc-index of at least n, but the weak c-closure is at most 3. $\square$

Remark. The bounds proven here can be visualized in a Hasse diagram (Figure 4.1). A line segment indicates that the lower parameter is bounded from above by some linear function of the upper parameter.

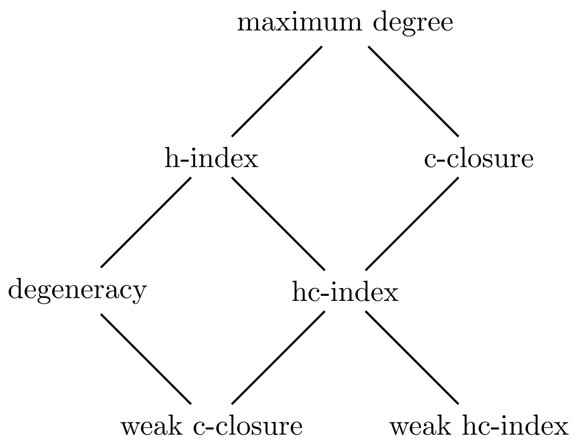

Figure 4.1: Hasse diagram of graph parameters.

Chapter 5

Computing the (weak) hc-index

The c-closure of a graph G can be computed efficiently: Algorithms with a running time of $\mathcal{O}(n_G^{2.373})$ [Fox+20] and $\mathcal{O}(\mathrm{c}(G) \cdot n_G^2 + m_G^{1.5})$ [KN21] are known. The former algorithm relies on fast matrix multiplication, whereas the latter is purely combinatorial.

The hc-index can be computed with a similar efficiency. We present a combinatorial algorithm with a running time of $\mathcal{O}(n_G + m_G \cdot \Delta(G))$ (Theorem 5.1), which was used to compute the hc-index of multiple real-world graphs (Table 5.1). Like the c-closure, the hc-index can be computed in $\mathcal{O}(n_G^{2.373})$ time using matrix multiplication (Theorem 5.2); however, for sparse graphs, the combinatorial algorithm is likely preferable.

The weak hc-index does not seem to be efficiently computable: Unless $P = NP$, the weak hc-index cannot be computed in polynomial time (Theorem 5.3). For this reason, no algorithm for computing the weak hc-index is presented here.

5.1 The hc-index of sparse graphs

In this section, an algorithm for computing the hc-index of sparse graphs is presented. It was used to compute the hc-index of several graphs from the real world.

Theorem 5.1. *The hc-index of a graph G can be computed in $\mathcal{O}(n_G + m_G \cdot \Delta(G))$ time.*

Proof. Using Algorithm 1, all closure numbers of a graph $G = (V, E)$ can be computed in $\mathcal{O}(n_G + m_G \cdot \Delta(G))$ time.

The algorithm calculates the closure number of $v \in V$ by counting the number of common neighbors between v and any vertex $u \in V$: The entry $A[u]$ is incremented if and only if a new common neighbor $w \in N_G(u) \cap N_G(v)$ is discovered. Afterwards, all entries in A of the vertices in $N_G[v]$ are reset to 0. Then, the algorithm searches for the maximum number of shared neighbors between v and any non-adjacent vertex $u \in V$ distinct from v. All entries of A need to be reset to 0 so A can be reused as auxiliary space for computing the next closure number.

For every vertex $v \in V$, Algorithm 1 iterates over $N_G(v)$ and takes $\mathcal{O}(\Delta(G))$ time for each $w \in N_G(v)$. By the degree sum formula, a call to AllClosureNumbers thus takes $\mathcal{O}(n_G + m_G \cdot \Delta(G))$ time in total.

To compute the hc-index of the graph G, the h-index of all closure numbers must be calculated. This can be done in $\mathcal{O}(n_G)$ time by first sorting the array of closure numbers in non-ascending order using *counting sort* [Cor+09, pp. 194–196], where the frequency of each closure number is counted, and then searching for the index $h \in \{1, 2, \ldots, n_G\}$ such that there are h closure numbers greater than or equal to h, while all other closure numbers are less than or equal to h. Clearly, $\mathrm{hc}(G) = h$. □

Algorithm 1 Algorithm to compute all closure numbers.

Input: A graph $G = (V, E)$ with $V = \{1, 2, \ldots, n_G\}$.
Output: All closure numbers of G.

```
A[n_G] ← ⟨0, 0, ..., 0⟩                ▷ Allocate and initialize an array of length n_G.

function ClosureNumber(G, v)
    for w ∈ N_G(v) do
        for u ∈ N_G(w) do
            A[u] ← A[u] + 1            ▷ Count the common neighbors between v and u.
        end for
    end for
    for u ∈ N_G[v] do
        A[u] ← 0                       ▷ Reset the irrelevant entries of A to 0.
    end for
    max ← 0
    for w ∈ N_G(v) do
        for u ∈ N_G(w) do
            if A[u] > max then
                max ← A[u]             ▷ Find the maximum number of common neighbors.
            end if
            A[u] ← 0                   ▷ Reset all the entries of A to 0.
        end for
    end for
    return max + 1
end function

function AllClosureNumbers(G)
    C[n_G] ← ⟨...⟩                     ▷ Allocate an array of length n_G.
    for v ← 1 to n_G do
        C[v] ← ClosureNumber(G, v)
    end for
    return C
end function
```

Remark. An implementation of Algorithm 1 was used to compute the c-closure and the hc-index of multiple real-world graphs. The results are shown in Table 5.1. For directed graphs, the underlying undirected graph was considered. Any loops were removed; thus, only simple graphs were analyzed. Note that the hc-index remains relatively small: Half of the selected graphs have an hc-index of at most 22.

Table 5.1: The maximum degree, h-index, c-closure, and hc-index of real-world graphs from the *Stanford Network Analysis Project* (SNAP) [LK14].

G	n_G	m_G	$\Delta(G)$	$\mathrm{h}(G)$	$\mathrm{c}(G)$	$\mathrm{hc}(G)$
ca-AstroPh	18,772	$2 \cdot 10^5$	504	150	61	47
ca-CondMat	23,133	93,439	279	76	27	20
ca-GrQc	5,242	14,484	81	45	43	25
ca-HepPh	12,008	$1.2 \cdot 10^5$	491	238	90	73
ca-HepTh	9,877	25,973	65	38	13	11
email-Enron	36,692	$1.8 \cdot 10^5$	1,383	195	187	80
p2p-Gnutella04	10,876	39,994	103	42	29	15
p2p-Gnutella05	8,846	31,839	88	46	48	25
p2p-Gnutella06	8,717	31,525	115	48	48	29
p2p-Gnutella08	6,301	20,777	97	51	31	23
p2p-Gnutella09	8,114	26,013	102	52	32	24
p2p-Gnutella24	26,518	65,369	355	33	11	9
p2p-Gnutella25	22,687	54,705	66	28	12	10
p2p-Gnutella30	36,682	88,328	55	35	18	12
p2p-Gnutella31	62,586	$1.5 \cdot 10^5$	95	41	18	13
roadNet-CA	$2 \cdot 10^6$	$2.8 \cdot 10^6$	12	8	5	4
roadNet-PA	$1.1 \cdot 10^6$	$1.5 \cdot 10^6$	9	8	4	4
roadNet-TX	$1.4 \cdot 10^6$	$1.9 \cdot 10^6$	12	8	4	4
wiki-Talk	$2.4 \cdot 10^6$	$4.7 \cdot 10^6$	$1 \cdot 10^5$	1,056	4,216	296
wiki-Vote	7,115	$1 \cdot 10^5$	1,065	186	441	113

5.2 The hc-index of dense graphs

For a dense graph, matrix multiplication can be used to compute the hc-index. Alman and Vassilevska Williams [AVW21] show that for every $n \in \mathbb{N}_1$, any two $n \times n$ matrices over a field can be multiplied using $\mathcal{O}(n^{\omega+\varepsilon})$ field operations, where $\omega \in \mathbb{R}$ denotes the *matrix multiplication exponent* with $2 \leq \omega < 2.3728596$, for all $\varepsilon \in \mathbb{R}_{>0}$. This gives the following upper bound on the complexity of hc-index computation:

Theorem 5.2. *The hc-index of an arbitrary graph G can be computed in $\mathcal{O}(n_G^{\omega+\varepsilon})$ time for any $\varepsilon \in \mathbb{R}_{>0}$.*

Proof. Fox et al. [Fox+20] mention that the c-closure of a graph G can be computed by squaring its adjacency matrix. This way, the number of common neighbors between any two vertices is obtained. To compute the hc-index, after squaring the adjacency matrix, we collect all closure numbers in $\mathcal{O}(n_G^2)$ time and calculate their h-index in $\mathcal{O}(n_G)$ time. Assuming that basic arithmetic operations take constant time, the total running time is bounded by $\mathcal{O}(n_G^{\omega+\varepsilon})$ for all $\varepsilon \in \mathbb{R}_{>0}$, because squaring the $n_G \times n_G$ adjacency matrix is the most expensive step. $\square$

5.3 The weak hc-index

Unlike the hc-index, the weak hc-index does not appear to be efficiently computable; in fact, the weak hc-index cannot be computed in polynomial time unless $P = NP$.

Theorem 5.3. *Given a graph G and a number $k \in \mathbb{N}_1$, deciding whether G has a weak hc-index of at most k is NP-complete.*

Proof. Clearly, the problem under consideration is in *NP*. We show that it is *NP*-hard as well. Let $G = (V, E)$ be a graph, and let $k \in \{2, 3, \dots, n_G - 1\}$ be a positive integer. Now, we construct a graph $G' := (V', E')$ as follows:

1. For each vertex $v \in V$, add v to V'.
2. For each edge $e \in E$, add $2 \cdot k$ new vertices $\{v_1^e, v_2^e, \dots, v_{2\cdot k}^e\}$ to V'.
3. For each edge $e = \{u, w\} \in E$, add the edges $\{\{u, v_i^e\}, \{v_i^e, w\} \mid i \in [2 \cdot k]\}$ to E'.

This can clearly be done in polynomial time. Then, G contains a vertex cover of size at most k if and only if G' has a weak hc-index of at most k, as we show next. Finally, the theorem follows directly from the *NP*-hardness of deciding if a vertex cover of at most a given size exists [GJ79, pp. 53–56].

($\Rightarrow$) Let $U \subseteq V$ with $|U| \leq k$ be a vertex cover in G. After removing the vertices in U from G', all closure numbers in the resulting graph are at most 2. Equivalently, we can state that $\mathrm{c}(G'[V' \setminus U]) \leq 2$. Since $2 \leq k \leq n_G - 1$, it follows that $\mathrm{whc}(G') \leq k$.

($\Leftarrow$) We will assume that $\mathrm{whc}(G') \leq k$: There exists a set $U \subsetneq V'$ with $|U| \leq k$ such that $\mathrm{c}(G'[V' \setminus U]) \leq k$. Now, there cannot be an edge $\{u, v\} \in E$ with $\{u, v\} \cap U = \varnothing$, as both u and v would have a closure number greater than k in $G'[V' \setminus U]$. Therefore, the set $W := U \cap V$ with $|W| \leq k$ is a vertex cover in G. $\square$

Chapter 6

Finding dominating sets

Parameterized by the solution size $k \in \mathbb{N}_1$, deciding whether a dominating set of size k exists in a graph is $W[2]$-complete and thus likely not fixed-parameter tractable [DF95]. However, fixed-parameter algorithms are known when combining the solution size with another parameter such as the c-closure [KKS20b] or the weak c-closure [LS21].

Koana et al. [KKS20b] show that given a graph G and a number $k \in \mathbb{N}_1$, deciding if the graph G contains a dominating set of size at most k takes $\mathcal{O}^*((\mathrm{c}(G) \cdot k)^{\mathcal{O}(k)})$ time. In this chapter, we adapt their algorithm so as to exploit the hc-index.

Lokshtanov and Surianarayanan [LS21] present an algorithm that exploits the weak c-closure: They show that given a graph G and a number $k \in \mathbb{N}_1$, deciding whether G contains a dominating set of size at most k takes $\mathcal{O}^*(k^{\mathcal{O}(\mathrm{wc}(G)^2 \cdot k^3)})$ time. The hc-index of a graph is greater than or equal to its weak c-closure, and so their algorithm is also suitable for graphs of small hc-index. However, we obtain a simpler algorithm for such graphs when adapting the algorithm by Koana et al. [KKS20b].

We follow their approach closely, and we use their definitions of the terms *bw-graph* and *bw-dominating set* [KKS20b, pp. 5, 12].

Definition 6.1. A *bw-graph* is a graph G that has the form $G = (B \uplus W, E)$, where B denotes the set of *black* vertices and W denotes the set of *white* vertices.

Definition 6.2. A set of vertices $D \subseteq V(G)$ in a bw-graph $G = (B \uplus W, E)$ is called a *bw-dominating set* if it dominates every black vertex, that is, $B \subseteq N_G[D]$.

At the end of this chapter, we present Algorithm 3. Given a graph G and a positive integer $k \in \mathbb{N}_1$, the algorithm decides if G contains a dominating set of size at most k in $\mathcal{O}^*((\mathrm{hc}(G) \cdot k)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(\mathrm{hc}(G))})$ time (Theorem 6.11).

The idea behind the algorithm is to try all suitable subsets of the vertices with high closure numbers as partial solutions. For each partial solution, we construct a bw-graph where the closed neighborhood of the partial solution is white and all other vertices are black. Then, we remove the vertices of the partial solution from the bw-graph. Further, we turn the remaining vertices of high closure number into a clique. Thus, the resulting bw-graph has bounded c-closure. However, we have to ensure that no vertices from the clique are included in a bw-dominating set for this bw-graph. To avoid this, we declare these vertices as *forbidden*. We modify the algorithm by Koana et al. [KKS20b] to deal with forbidden vertices; for sufficiently small instances, Algorithm 2 is invoked.

Formally, we first study the problem below.

Constrained BW-Dominating Set
Input: A bw-graph $G = (B \uplus W, E)$, a solution size $k \in \mathbb{N}_0$, and a set of forbidden vertices $F \subseteq V(G)$.
Question: Is there a set $D \subseteq V(G) \setminus F$ with $|D| \leq k$ such that $B \subseteq N_G[D]$?

Following the approach by Koana et al. [KKS20b, p. 8], we can use a reduction rule to reduce the number of white vertices in an instance of the problem.

Reduction Rule 6.3. *Given $G = (B \uplus W, E)$ and $F \subseteq V(G)$, if there are two distinct vertices $u \in W$ and $v \in V(G) \setminus F$ with $N_G(u) \cap B \subseteq N_G[v] \cap B$, then remove u.*

It can clearly be applied exhaustively in polynomial time and does not increase the c-closure, so we move on to proving its correctness.

Lemma 6.4. *Reduction Rule 6.3 is correct.*

Proof. Let $G = (B \uplus W, E)$ be a bw-graph, let $F \subseteq V(G)$ be a set of forbidden vertices, and let $G' = (B \uplus W', E')$ be the bw-graph obtained after applying the reduction rule to the vertices $u \in W \setminus W'$ and $v \in V(G') \setminus F$. Clearly, any bw-dominating set in G' is also a bw-dominating set in G. In addition, any bw-dominating set in G that does not contain u is a bw-dominating set in G'. Finally, if $D \subseteq V(G) \setminus F$ with $u \in D$ is a bw-dominating set in G, then $D' := (D \setminus \{u\}) \cup \{v\}$ is a bw-dominating set in G' such that $D' \subseteq V(G') \setminus F$ and $|D'| \leq |D|$, so the reduction rule is correct. □

Using the reduction rule, an algorithm for Constrained BW-Dominating Set is obtained that is suitable for instances with few black and few forbidden vertices. It is used to solve such instances in Algorithm 3.

Algorithm 2 Algorithm for Constrained BW-Dominating Set.

Input: A bw-graph $G = (B \uplus W, E)$, a number $k \in \mathbb{N}_0$, and a set $F \subseteq V(G)$.
Output: If G contains a Constrained BW-Dominating Set, then `YES`, else `NO`.

function SolveCBWDS($G = (B \uplus W, E), k, F$)
  Apply Reduction Rule 6.3 exhaustively to (G, F).
  $\overline{F} \leftarrow V(G) \setminus F$
  $P \leftarrow \{v \in \overline{F} \mid N_G[v] \cap B \cap \overline{F} \neq \varnothing\}$
  for $D_* \subseteq P$ **with** $|D_*| \leq k$ **do**
    if $B \cap \overline{F} \subseteq N_G[D_*]$ **then**
      $U \leftarrow (B \cap F) \setminus N_G[D_*]$
      $S \leftarrow \{N_G[v] \cap U \mid v \in \overline{F} \setminus P\}$
      if SolveSC$(U, S, k - |D_*|) =$ `YES` **then** ▷ Reduction to Set Cover.
        return `YES`
      end if
    end if
  end for
  return `NO`
end function

The algorithm invokes the SolveSC function to solve instances of the Set Cover problem, where it must be decided whether there are $k \in \mathbb{N}_0$ or fewer sets in $S \subseteq \mathcal{P}(U)$ such that their union is equal to the set U, which can be done in $\mathcal{O}(|U| \cdot |S| \cdot 2^{|U|})$ time using the algorithm by Fomin and Kratsch [FK10, p. 36]. We refer to this bound in the analysis of Algorithm 2 below.

Lemma 6.5. *Algorithm 2 is correct.*

Proof. Let $G = (B \uplus W, E)$ be an arbitrary bw-graph, let $k \in \mathbb{N}_0$ be a solution size, and let $F \subseteq V(G)$ be a set of forbidden vertices. We show that G contains a bw-dominating set that satisfies these constraints if and only if Algorithm 2 returns `YES`. Henceforth, let $\overline{F} = V(G) \setminus F$ be the set that is constructed in Line 3, and let $P \subseteq \overline{F}$ be the set that is constructed in Line 4.

($\Rightarrow$) Let $D \subseteq \overline{F}$ with $|D| \leq k$ be a set such that $B \subseteq N_G[D]$. Choose $D_* := D \cap P$. Clearly, we have $|D_*| \leq k$ and $B \cap \overline{F} \subseteq N_G[D_*]$. Now, let $U = (B \cap F) \setminus N_G[D_*]$ be the set constructed in Line 7, and let $S \subseteq \mathcal{P}(U)$ be the set constructed in Line 8. We show that $(U, S, k - |D_*|)$ is a `YES`-instance of Set Cover: Choose $D' := D \setminus D_*$. Since we have $U \subseteq N_G[D']$, the union over $S' := \{N_G[v] \cap U \mid v \in D'\}$ equals U. From $S' \subseteq S$ and $|S'| \leq k - |D_*|$, we conclude that Algorithm 2 returns `YES`.

($\Leftarrow$) We will assume that Algorithm 2 returns `YES`. Then, there exists a set $D_* \subseteq P$ with $|D_*| \leq k$ such that $B \cap \overline{F} \subseteq N_G[D_*]$. Let $U = (B \cap F) \setminus N_G[D_*]$ be the set that is constructed in Line 7, and let $S \subseteq \mathcal{P}(U)$ be the set that is constructed in Line 8. As there is a set $S' \subseteq S$ with $|S'| \leq k - |D_*|$ such that the union over S' equals U, there is a set $D' \subseteq \overline{F} \setminus P$ with $|D'| \leq k - |D_*|$ and $U \subseteq N_G[D']$. Choose $D := D' \cup D_*$. We conclude that the set $D \subseteq \overline{F}$ with $|D| \leq k$ is such that $B \subseteq N_G[D]$. □

Lemma 6.6. *Given any bw-graph $G = (B \uplus W, E)$, number $k \in \mathbb{N}_0$, and set $F \subseteq V(G)$, the running time of Algorithm 2 is bounded by $\mathcal{O}^*\big((\mathrm{c}(G) \cdot |B|)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(|F|)}\big)$.*

Proof. Let $G = (B \uplus W, E)$ be an arbitrary bw-graph, let $k \in \mathbb{N}_0$ be a solution size, and let $F \subseteq V(G)$ be a set of forbidden vertices. Furthermore, let $\overline{F} = V(G) \setminus F$ be the set that is constructed in Line 3, and let $P \subseteq \overline{F}$ be the set that is constructed in Line 4. As Reduction Rule 6.3 has been applied exhaustively, the black neighborhood of any white vertex in P cannot be a clique; otherwise, this white vertex would have been removed. Thus, every white vertex in P has two black neighbors that are not adjacent. Then, by the definition of c-closure, the set P contains $\mathcal{O}(\mathrm{c}(G) \cdot |B|^2)$ vertices in total. The loop in Line 5 therefore iterates over $\mathcal{O}((\mathrm{c}(G) \cdot |B|)^{\mathcal{O}(k)})$ subsets $D_* \subseteq P$. Finally, after the set $U \subseteq F$ is constructed in Line 7 and the set $S \subseteq \mathcal{P}(U)$ is constructed in Line 8, the resulting Set Cover instance $(U, S, k - |D_*|)$ is solved in $\mathcal{O}(|F| \cdot 2^{|F|} \cdot 2^{|F|})$ time using the algorithm by Fomin and Kratsch [FK10, p. 36]. Consequently, the running time of Algorithm 2 is bounded by $\mathcal{O}^*((\mathrm{c}(G) \cdot |B|)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(|F|)})$. □

In the algorithm for finding dominating sets (Algorithm 3), the following reduction rule gives us a bound on the number of black neighbors of any black vertex.

Reduction Rule 6.7. *Given $G = (B \uplus W, E)$ and $k \in \mathbb{N}_0$, if a black vertex $v \in B$ has more than $\mathrm{c}(G) \cdot k$ black neighbors, then color v white.*

Again, the reduction rule can be applied exhaustively in polynomial time, and so it remains to prove its correctness.

Lemma 6.8. *Reduction Rule 6.7 is correct.*

Proof. Let $G = (B \uplus W, E)$ be any bw-graph, let $k \in \mathbb{N}_0$ be an arbitrary solution size, and let $G' = (B' \uplus W', E)$ be the bw-graph obtained by applying Reduction Rule 6.7 to the vertex $v \in B \cap W'$. Furthermore, let $F \subseteq V(G)$ be a set of forbidden vertices. We show that any set $D \subseteq V(G) \setminus F$ with $|D| \leq k$ is a bw-dominating set in G if and only if it is a bw-dominating set in G'.

($\Rightarrow$) Let $D \subseteq V(G) \setminus F$ with $|D| \leq k$ be a bw-dominating set in G. Then, $B \subseteq N_G[D]$. Because $B' \subseteq B$ and $N_{G'}[D] = N_G[D]$, it follows that $B' \subseteq N_{G'}[D]$. Therefore, D is also a bw-dominating set in G'.

($\Leftarrow$) Let $D \subseteq V(G') \setminus F$ with $|D| \leq k$ be a bw-dominating set in G'. We prove that D must contain v or at least one of its neighbors: Assume, for the sake of contradiction, that $N_{G'}[v] \cap D = \varnothing$. By the definition of c-closure, every vertex in D can dominate at most $\mathrm{c}(G') - 1$ black neighbors of v. Thus, all the vertices in D combined dominate at most $(\mathrm{c}(G') - 1) \cdot k \leq \mathrm{c}(G') \cdot k$ black neighbors of v. Since v has more than $\mathrm{c}(G') \cdot k$ black neighbors, this is in contradiction to D being a bw-dominating set in G'. Hence, we conclude that $N_{G'}[v] \cap D \neq \varnothing$, and so D dominates v. As $B = B' \cup \{v\}$, it follows that D is a bw-dominating set in G. □

Next, we turn to the algorithm for finding dominating sets; we prove its correctness and analyze its running time.

Lemma 6.9. *Algorithm 3 is correct.*

Proof. Let $G = (V, E)$ be a graph, let $k \in \mathbb{N}_1$ be a solution size, and let $H \subseteq V$ be the set that is constructed in Line 2. We show that G contains a dominating set of size at most k if and only if Algorithm 3 returns `YES`.

($\Rightarrow$) Let $D \subseteq V$ with $|D| \leq k$ be a dominating set in G. Choose $D_* := D \cap H$. Now, let $F = H \setminus D_*$ be the set that is constructed in Line 6. Also, let $G' = (B \uplus W, E')$ be the bw-graph that is constructed in Line 8. Then, $(G', k - |D_*|, F)$ is a `YES`-instance of Constrained BW-Dominating Set: Choose $D' := D \setminus D_*$. Clearly, $D' \subseteq V(G') \setminus F$ and $|D'| \leq k - |D_*|$. Because $B = V \setminus N_G[D_*]$, it follows that $B \subseteq N_G[D']$. And hence, from $E' = (E \cap \binom{V \setminus D_*}{2}) \cup \binom{F}{2}$, we conclude that $B \subseteq N_{G'}[D']$. Finally, since Branch works analogously to *Branch* [KKS20b, p. 10], it will recognize the `YES`-instance. Thus, Algorithm 3 returns `YES`.

($\Leftarrow$) We will assume that Algorithm 3 returns `YES`. Then, there exists a set $D_* \subseteq H$ with $|D_*| \leq k$ such that Branch returns `YES` on $(G', k - |D_*|, F)$, where $F = H \setminus D_*$ is the set and $G' = (B \uplus W, E')$ the bw-graph that is constructed in Line 6 and Line 8, respectively. Because Branch works analogously to *Branch* [KKS20b, p. 10], we infer that $(G', k - |D_*|, F)$ is a `YES`-instance of Constrained BW-Dominating Set, and thus there is a set $D' \subseteq V(G') \setminus F$ with $|D'| \leq k - |D_*|$ such that $B \subseteq N_{G'}[D']$. Since we have $E' = (E \cap \binom{V \setminus D_*}{2}) \cup \binom{F}{2}$, it follows that $B \subseteq N_G[D']$. Choose $D := D' \cup D_*$. From $B = V \setminus N_G[D_*]$, we deduce that D with $|D| \leq k$ is a dominating set in G, that is, $V \subseteq N_G[D]$. □

Algorithm 3 Algorithm to find a dominating set, based on *SolveTDS* [KKS20b, p. 10].

Input: A graph $G = (V, E)$ and a number $k \in \mathbb{N}_1$.
Output: If G contains a dominating set of size at most k, then `YES`, else `NO`.

function SolveDS($G = (V, E), k$)
    $H \leftarrow \{v \in V \mid \mathrm{cl}_G(v) > \mathrm{hc}(G)\}$
    for $D_* \subseteq H$ **with** $|D_*| \leq k$ **do**
        $B \leftarrow V \setminus N_G[D_*]$
        $W \leftarrow N_G(D_*)$
        $F \leftarrow H \setminus D_*$
        $E' \leftarrow (E \cap \binom{V \setminus D_*}{2}) \cup \binom{F}{2}$
        $G' \leftarrow (B \uplus W, E')$
        Apply Reduction Rule 6.7 exhaustively to $(G', k - |D_*|)$.
        if Branch($G', k - |D_*|, F, \varnothing$) $=$ `YES` **then**
            return `YES` ▷ Success.
        end if
    end for
    return `NO` ▷ Failure.
end function

function Branch($G = (B \uplus W, E), k, F, D$)
    Color $N_G[D]$ white.
    if $B = \varnothing$ **then**
        return `YES`
    end if
    if $k = 0$ **then**
        return `NO`
    end if
    Find a maximal independent set $I \subseteq B$ in $G[B]$.
    if $|I| \geq k + 1$ **then**
        Pick an arbitrary set $I' \subseteq I$ with $|I'| = k + 1$.
        $P \leftarrow \{v \in V(G) \mid |N_G(v) \cap I'| \geq 2\}$
        for $v \in P \setminus (F \cup D)$ **do**
            if Branch($G, k - 1, F, D \cup \{v\}$) $=$ `YES` **then**
                return `YES`
            end if
        end for
        return `NO`
    end if
    return SolveCBWDS(G, k, F)
end function

It remains to analyze the running time of the algorithm.

Lemma 6.10. *Given a graph G and a number $k \in \mathbb{N}_1$, the running time of Algorithm 3 is bounded by $\mathcal{O}^*\big((\mathrm{hc}(G) \cdot k)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(\mathrm{hc}(G))}\big)$.*

Proof. Let $G = (V, E)$ be a graph, let $k \in \mathbb{N}_1$ be a solution size, and let $H \subseteq V$ be the set that is constructed in Line 2. Because $|H| \leq \mathrm{hc}(G)$, the for-loop in Line 3 iterates over $\mathcal{O}(\mathrm{hc}(G)^k)$ subsets $D_* \subseteq H$. Then, let $F = H \setminus D_*$ be the set that is constructed in Line 6, and let $G' = (B \uplus W, E')$ be the bw-graph that is constructed in Line 8. We have $|F| \leq \mathrm{hc}(G)$ since $F \subseteq H$.

Additionally, we have $\mathrm{c}(G') \leq 2 \cdot \mathrm{hc}(G)$: Let $u, v \in V(G')$ be two distinct vertices that are non-adjacent in G'. At most one of the two vertices can be in H because any two distinct vertices from $F = H \cap V(G')$ are adjacent in G'. Thus, u and v share at most $\mathrm{hc}(G) - 1$ neighbors in G. As $E' = (E \cap \binom{V \setminus D_*}{2}) \cup \binom{F}{2}$, if one of the two vertices is in F, then it will share at most $\mathrm{hc}(G) + |F| - 1 \leq 2 \cdot \mathrm{hc}(G) - 1$ neighbors with the respective other vertex in G'. Otherwise, they also share at most $\mathrm{hc}(G) - 1$ neighbors in the bw-graph G'. Therefore, $\mathrm{c}(G') \leq 2 \cdot \mathrm{hc}(G)$.

After the exhaustive application of Reduction Rule 6.7 in Line 9, any black vertex in G' has at most $\mathrm{c}(G') \cdot k \leq 2 \cdot \mathrm{hc}(G) \cdot k$ black neighbors. As the number of calls to Branch is bounded by $\mathcal{O}((\mathrm{c}(G') \cdot k)^{\mathcal{O}(k)}) \subseteq \mathcal{O}((\mathrm{hc}(G) \cdot k)^{\mathcal{O}(k)})$ [KKS20b, p. 11], only the running time of Line 36 remains to be bounded.

In Line 36, the independent set $I \subseteq B$ constructed in Line 25 is such that $|I| \leq k$ and $B \subseteq N_{G'}[I]$, so $|B| = |N_{G'}[I] \cap B| \leq 2 \cdot \mathrm{hc}(G) \cdot k^2 + k$. By Lemma 6.6, it follows that Line 36 takes $\mathcal{O}^*((\mathrm{hc}(G) \cdot k)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(\mathrm{hc}(G))})$ time. Combining all the bounds, the total running time of Algorithm 3 is also $\mathcal{O}^*((\mathrm{hc}(G) \cdot k)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(\mathrm{hc}(G))})$. □

Finally, from Lemma 6.9 and Lemma 6.10, we obtain the following:

Theorem 6.11. *Given a graph G and a number $k \in \mathbb{N}_1$, deciding whether G contains a dominating set of size at most k takes $\mathcal{O}^*\big((\mathrm{hc}(G) \cdot k)^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(\mathrm{hc}(G))}\big)$ time.*

Chapter 7

Counting maximal cliques

Arbitrary graphs can contain many maximal cliques: For every $n \in \mathbb{N}_1$, there is a graph on n vertices with $3^{\lfloor n/3 \rfloor}$ maximal cliques [MM65]. However, graphs with small c-closure contain relatively few maximal cliques. Fox et al. [Fox+20] show that for all $n, c \in \mathbb{N}_1$, a graph on n vertices with a c-closure of at most c contains at most $4^{(c+4)\cdot(c-1)/2} \cdot n^{2-2^{1-c}}$ maximal cliques. They also prove that for all $n, c \in \mathbb{N}_1$, every graph with n vertices and a weak c-closure of at most c contains at most $3^{(c-1)/3} \cdot n^2$ maximal cliques. The goal of this chapter is to derive similar bounds for the weak hc-index.

It is useful to prove an upper bound on the number of maximal cliques that exploits the weak hc-index since listing all maximal cliques in a graph G takes $\mathcal{O}(n_G \cdot m_G)$ time per maximal clique [Tsu+77]. Therefore, if graphs of small weak hc-index contain only few maximal cliques, then these maximal cliques can be listed quickly. In particular, by adapting the bounds from Fox et al. [Fox+20], it follows that the problem of generating all maximal cliques is fixed-parameter tractable in the weak hc-index.

To adapt the bounds, we show that the number of maximal cliques in a graph with a given weak hc-index is bounded by the number of maximal cliques in certain induced subgraphs of small c-closure (Theorem 7.4). The new bound maintains the dependence on the number of vertices, and thus we obtain Corollary 7.5, adapting the subquadratic bound by Fox et al. [Fox+20].

To start with, we state their definition of the function F [Fox+20, p. 453].

Definition 7.1. For all $n, c \in \mathbb{N}_1$, we denote the maximum number of maximal cliques in a graph on n vertices that has c-closure at most c as $F(n, c) \in \mathbb{N}_1$.

Fox et al. [Fox+20] show that $F(n, c) \leq \min\{3^{(c-1)/3} \cdot n^2, 4^{(c+4)\cdot(c-1)/2} \cdot n^{2-2^{1-c}}\}$ for all $n, c \in \mathbb{N}_1$. Next, we define a function C, and we prove a bound on C using F. Then, their bound on F gives us a bound on C.

Definition 7.2. For all $n, k \in \mathbb{N}_1$, we denote the maximum number of maximal cliques in a graph on n vertices with a weak hc-index of k as $C(n, k) \in \mathbb{N}_0$, where $C(n, k) \coloneqq 0$ if no such graph exists.

To simplify the subsequent proof, we use the following notation.

Definition 7.3. For a graph $G = (V, E)$ and a non-empty set $U \subseteq V$, we define:

$$N_G^{\cap}(U) \coloneqq \bigcap_{v \in U} N_G(v)$$

Now, we obtain the bound on the maximum number of maximal cliques. It applies to every graph G with $n_G > \mathrm{whc}(G)$; we only exclude graphs isomorphic to K_1.

Theorem 7.4. *For all $n, k \in \mathbb{N}_1$ with $n > k$, the bound $C(n, k) \leq 2^k \cdot F(n-k, k)$ holds.*

Proof. Let $G = (V, E)$ be a graph with $n_G > 1$ and $C(n_G, \mathrm{whc}(G))$ maximal cliques. By definition of the weak hc-index, there is a set of vertices $U \subsetneq V$ with $|U| = \mathrm{whc}(G)$ such that $\mathrm{c}(G[W]) \leq \mathrm{whc}(G)$, where $W := V \setminus U$. Since $|W| = n_G - \mathrm{whc}(G)$, the number of maximal cliques in $G[W]$ is at most $F(n_G - \mathrm{whc}(G), \mathrm{c}(G[W]))$. Clearly, this is bounded from above by $F(n_G - \mathrm{whc}(G), \mathrm{whc}(G))$.

It remains to bound the maximal cliques that have at least one vertex in U. Such maximal cliques either lie entirely in U or can be partitioned into a clique $X \subseteq U$ and a maximal clique in $G[N_G^{\cap}(X) \cap W]$: No maximal clique in G consists of a clique $X \subseteq U$ and some non-maximal clique in $G[N_G^{\cap}(X) \cap W]$, because any vertex that extends the non-maximal clique in $G[N_G^{\cap}(X) \cap W]$ would extend the maximal clique in G, which is impossible.

There are exactly $2^{\mathrm{whc}(G)} - 1$ non-empty subsets $X \subseteq U$. For every such set X, we have $|N_G^{\cap}(X) \cap W| \leq n_G - \mathrm{whc}(G)$ and $\mathrm{c}(G[N_G^{\cap}(X) \cap W]) \leq \mathrm{whc}(G)$ if $N_G^{\cap}(X) \cap W$ is not empty. Hence, the number of maximal cliques that have at least one vertex in U is at most $(2^{\mathrm{whc}(G)} - 1) \cdot F(n_G - \mathrm{whc}(G), \mathrm{whc}(G))$. Combining the bounds, it follows that there are at most $2^{\mathrm{whc}(G)} \cdot F(n_G - \mathrm{whc}(G), \mathrm{whc}(G))$ maximal cliques in G, and thus we have $C(n_G, \mathrm{whc}(G)) \leq 2^{\mathrm{whc}(G)} \cdot F(n_G - \mathrm{whc}(G), \mathrm{whc}(G))$. $\square$

Using the bound by Fox et al. [Fox+20], we conclude the following:

Corollary 7.5. *For any fixed $k \in \mathbb{N}_1$, every graph G with a weak hc-index of at most k contains $\mathcal{O}\big(n_G^{2-2^{1-k}}\big)$ maximal cliques.*

Chapter 8

Conclusion

In this thesis, we explored algorithmic as well as structural aspects of the hc-index and the weak hc-index. We showed how the graphs of small hc-index can be characterized in terms of forbidden induced subgraphs. Then, we determined the position of the hc-index and the weak hc-index in the graph parameter hierarchy. Next, we presented a simple algorithm that efficiently computes the hc-index of a graph. For the weak hc-index, we showed that it is unlikely to be computable in polynomial time. In the subsequent two chapters, we used the hc-index and the weak hc-index to adapt results that rely on the c-closure. In particular, we adapted an algorithm by Koana et al. [KKS20b] for finding dominating sets so as to exploit the hc-index. Further, we showed that the bounds for the maximum number of maximal cliques by Fox et al. [Fox+20] can be adapted to obtain similar bounds for the weak hc-index. Consequently, the maximum number of maximal cliques in graphs of bounded weak hc-index is subquadratic in the number of vertices, and the problem of generating all maximal cliques in a graph is fixed-parameter tractable with respect to the weak hc-index.

These results suggest that the (weak) hc-index is generally useful for adapting results that depend on the c-closure. As such, the parameters enable gradual progress: Results that rely on the c-closure may be adapted to use the hc-index instead. The new results could then be adapted to use the weak c-closure or the weak hc-index, moving down the graph parameter hierarchy.

The thesis mainly focused on the hc-index; the weak hc-index was not investigated to the same extent. Future research could, for instance, characterize the graphs of small weak hc-index. It could also be attempted to prove more bounds on the weak hc-index. While the weak hc-index does not appear to be efficiently computable for general graphs, there may be efficient algorithms for special graph classes. This could be of interest when using the weak hc-index for the design of parameterized algorithms. For example, can the weak hc-index be used for finding dominating sets?

Finally, I want to thank Tomohiro Koana and Dr. André Nichterlein for supervising this thesis, and I want to thank the Studienstiftung des deutschen Volkes for supporting my studies in computer science.

Literature

[AVW21] Josh Alman and Virginia Vassilevska Williams. *A Refined Laser Method and Faster Matrix Multiplication*. In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA 2021)*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2021, pp. 522–539 (cit. on p. 29).

[BM08] John A. Bondy and U. S. R. Murty. *Graph Theory*. Vol. 244. Graduate Texts in Mathematics. London, England: Springer-Verlag, 2008. Repr. (cit. on p. 15).

[Cor+09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. 3rd ed. Cambridge, MA: The MIT Press, 2009 (cit. on p. 28).

[DF95] Rodney G. Downey and Michael R. Fellows. *Fixed-Parameter Tractability and Completeness I: Basic Results*. In: *SIAM Journal on Computing* 24.4 (1995), pp. 873–921 (cit. on pp. 14, 31).

[Die00] Reinhard Diestel. *Graph Theory*. 2nd ed. Vol. 173. Graduate Texts in Mathematics. New York, NY: Springer-Verlag, 2000 (cit. on p. 15).

[ELS10] David Eppstein, Maarten Löffler, and Darren Strash. *Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time*. In: *Algorithms and Computation (ISAAC 2010)*. Vol. 6506. Lecture Notes in Computer Science. Berlin; Heidelberg: Springer-Verlag, 2010, pp. 403–414 (cit. on p. 14).

[ES09] David Eppstein and Emma S. Spiro. *The h-Index of a Graph and Its Application to Dynamic Subgraph Statistics*. In: *Algorithms and Data Structures (WADS 2009)*. Vol. 5664. Lecture Notes in Computer Science. Berlin; Heidelberg: Springer-Verlag, 2009, pp. 278–289 (cit. on pp. 14, 17).

[Esc+11] Elaine M. Eschen, Chính T. Hoàng, Jeremy P. Spinrad, and R. Sritharan. *On graphs without a C_4 or a diamond*. In: *Discrete Applied Mathematics* 159.7 (2011), pp. 581–587 (cit. on p. 19).

[FK10] Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Berlin; Heidelberg: Springer-Verlag, 2010 (cit. on p. 33).

[Fox+20] Jacob Fox, Tim Roughgarden, C. Seshadhri, Fan Wei, and Nicole Wein. *Finding Cliques in Social Networks: A New Distribution-Free Model*. In: *SIAM Journal on Computing* 49.2 (2020), pp. 448–464 (cit. on pp. 11–14, 16, 19, 23, 27, 29, 37–39).

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. A Series of Books in the Mathematical Sciences. New York, NY: W. H. Freeman and Company, 1979 (cit. on p. 30).

[Gra73] Mark S. Granovetter. *The Strength of Weak Ties*. In: *American Journal of Sociology* 78.6 (1973), pp. 1360–1380 (cit. on p. 11).

[Hir05] Jorge E. Hirsch. *An index to quantify an individual's scientific research output*. In: *Proceedings of the National Academy of Sciences* 102.46 (2005), pp. 16569–16572 (cit. on pp. 13, 17).

[KKS20a] Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. *Computing Dense and Sparse Subgraphs of Weakly Closed Graphs*. In: *31st International Symposium on Algorithms and Computation (ISAAC 2020)*. Vol. 181. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 20:1–20:17 (cit. on pp. 11, 12, 16, 19, 23).

[KKS20b] Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. *Exploiting c-Closure in Kernelization Algorithms for Graph Problems*. In: *28th Annual European Symposium on Algorithms (ESA 2020)*. Vol. 173. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 65:1–65:17 (cit. on pp. 11–14, 23, 31, 32, 34–36, 39).

[KN21] Tomohiro Koana and André Nichterlein. *Detecting and enumerating small induced subgraphs in c-closed graphs*. In: *Discrete Applied Mathematics* 302 (2021), pp. 198–207 (cit. on pp. 23, 27).

[Koa+22] Tomohiro Koana, Christian Komusiewicz, André Nichterlein, and Frank Sommer. *Covering Many (Or Few) Edges with k Vertices in Sparse Graphs*. In: *39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)*. Vol. 219. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 42:1–42:18 (cit. on p. 23).

[LK14] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. June 2014. URL: `https://snap.stanford.edu/data` (visited on 06/06/2022) (cit. on p. 29).

[LS21] Daniel Lokshtanov and Vaishali Surianarayanan. *Dominating Set in Weakly Closed Graphs is Fixed Parameter Tractable*. In: *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*. Vol. 213. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 29:1–29:17 (cit. on pp. 12, 14, 31).

[LW70] Don R. Lick and Arthur T. White. *k-Degenerate Graphs*. In: *Canadian Journal of Mathematics* 22.5 (1970), 1082–1096 (cit. on pp. 14, 16).

[MM65] John W. Moon and Leo Moser. *On cliques in graphs*. In: *Israel Journal of Mathematics* 3.1 (1965), pp. 23–28 (cit. on pp. 12, 14, 37).

[Nie06] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications 31. Oxford, England: Oxford University Press, 2006 (cit. on p. 15).

[SW19] Manuel Sorge and Mathias Weller. *The Graph Parameter Hierarchy*. Feb. 2019. URL: `https://manyu.pro/assets/parameter-hierarchy.pdf` (visited on 08/18/2022) (cit. on p. 23).

[Tsu+77] Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. *A New Algorithm for Generating All the Maximal Independent Sets*. In: *SIAM Journal on Computing* 6.3 (1977), pp. 505–517 (cit. on p. 37).