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

Placing Green Bridges to Reconnect Habitats Densely: Algorithms and Complexity

Christian Wallisch

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

April 2023

First reviewer: Prof. Dr. Mathias Weller
Second reviewer: Prof. Dr. Stefan Schmid
Supervisors: Dr. Till Fluschnik and Leon Kellerhals

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, den ____________________ ____________________________

Datum Unterschrift

Abstract

We consider the approach to wildlife corridor design presented by Fluschnik and Kellerhals [CiE '21]. They utilize graph theory to model fragmented habitats. In their framework, vertices represent patches of land separated by man-made barriers such as motorways. Edges represent potential locations for wildlife crossings, also called green bridges, spanning the barriers. Additionally, their approach takes into account that the habitats of different species may encompass different subsets of the disconnected land areas. Given a cost budget and a connectivity requirement, the objective is to select places to build wildlife crossings such that the habitat of each species is reconnected in a way that fulfills the connectivity requirement.

We focus on connectivity requirements enforcing dense solutions in the sense that animals can reach any area of their habitat by crossing a small number of green bridges. In particular, we study the 2-Diam GBP problem introduced by Fluschnik and Kellerhals [CiE '21]. We identify special cases that permit exact polynomial-time algorithms, as well as NP-hard special cases. We pay particular attention to cases where the graph is planar, has low maximum degree, or where all habitats are small. As a byproduct, we obtain positive answers for two questions regarding the related 1-Reach GBP problem left open in a paper by Herkenrath et al. [IJCAI '22]. Moreover, we contribute to the understanding of the parameterized complexity of 2-Diam GBP. Lastly, we introduce $(2,2)$-Closed GBP, a variant of 2-Diam GBP, and conduct a similar analysis as for 2-Diam GBP.

Zusammenfassung

Wir befassen uns mit dem von Fluschnik und Kellerhals [CiE '21] vorgestellten Ansatz zur Gestaltung von Wildtierkorridoren. Die genannten Autoren bedienen sich der Graphentheorie, um fragmentierte Habitate zu modellieren. In ihrem Modell stellen die Knoten Landstriche dar, die durch künstliche Barrieren wie z. B. Autobahnen getrennt sind. Die Kanten stellen mögliche Standorte für Grünbrücken dar, welche die künstlichen Barrieren überspannen. Darüber hinaus berücksichtigt der Ansatz, dass Habitate verschiedener Tierarten unterschiedliche Teilmengen der getrennten Landstriche umfassen können. Bei einem vorgegebenem Kostenbudget und einer Verknüpfungsanforderung besteht das Ziel darin, Orte für den Bau von Grünbrücken auszuwählen, sodass der Lebensraum jeder Tierart unter Berücksichtigung der Verknüpfungsanforderung wieder verbunden ist.

Wir konzentrieren uns auf Verknüpfungsanforderungen, die dichte Lösungen in dem Sinne vorschreiben, dass Tiere jeden Bereich ihres Lebensraums über eine geringe Zahl von Grünbrücken erreichen können. Insbesondere untersuchen wir das von Fluschnik und Kellerhals [CiE '21] eingeführte 2-Diam GBP-Problem. Wir identifizieren Spezialfälle,

die exakte Polynomialzeitalgorithmen zulassen, sowie NP-schwere Spezialfälle. Besonderes Augenmerk richten wir auf Fälle, in denen der zugrundeliegende Graph planar ist, einen niedrigen Maximalgrad hat oder in denen alle Habitate von geringer Größe sind. Als ein Nebenergebnis erhalten wir positive Antworten auf zwei Fragen zum verwandten 1-Reach GBP-Problem, die in einer Arbeit von Herkenrath u. a. [IJCAI '22] offen gelassen wurden. Des Weiteren tragen wir zum Verständnis der parametrisierten Komplexität von 2-Diam GBP bei. Zuletzt führen wir mit $(2,2)$-Closed GBP eine Variante von 2-Diam GBP ein und betreiben eine ähnliche Analyse wie für 2-Diam GBP.

Contents

Chapter 1

Introduction

Lias [KW21], a creature of pride and strength, dwells within lush and vibrant forests, where he mostly remains hidden from the oblivious eyes of humans. But when glimpsed, he is a sight to behold, with his luxuriant fur and sleek muscular frame. His keen senses and sharp teeth make him the unrivaled apex predator of his expansive territory along the Danube river. He is not just any feline, but a lynx, one of the most unlikely animals to be encountered in Germany due to its masterful stealth and great rarity.

He is no ordinary lynx either, for he undertook a long and treacherous journey from his birthplace in the Swiss Jura Mountains, crossing many obstacles, including hazardous motorways, to reach his current domain, where no lynx has lived for over a century. Yet, Lias is faced with heartache and even humans have noticed. His vast kingdom seems eerily empty, as far and wide there are no female lynxes, who tend to be much more hesitant about crossing obstacles [DPL21].

To help Lias and animals of many different species in all kinds of predicaments caused by man-made obstacles, measures to increase the permeability of these obstacles need to be taken. The greater aim of this is to preserve endangered species, prevent genetic decline caused by inbreeding, restore biodiversity, facilitate seasonal animal migration, and aid animals in adapting to climate change [Ben03; CK15; HZ09]. Lynxes in particular are classified as critically endangered by the German Federal Agency for Nature Conservation, with habitat fragmentation explicitly cited as a cause [Mei+20].

Dedicated overpasses, underpasses, and a variety of further structures like nets and poles allow animals to cross streets more easily and safely [SRR15], which also contributes to protecting humans from wildlife-vehicle collisions [MVM10]. Simplifying, we refer to all of these structures as green bridges. A critical property in the design of wildlife corridors is the distance between habitat patches and the number of obstacles that need to be crossed between them [Bro+15; New93]. Hence, it seems desirable to place green bridges such that every animal can reach any part of its habitat by traversing only a few green bridges, resulting in densely connected habitats. A natural question is where to build green bridges to achieve dense connection of habitats while keeping financial costs low.

A framework to tackle this question has been developed by Fluschnik and Kellerhals [FK21]. They define three classes of computational problems, each representing a different demand on habitat connectivity. Algorithms for these problems can be used to compute optimal locations for green bridges under the respective connectivity constraint. To obtain the best results possible even when considering heavily fragmented habitats of various species at once, yielding a myriad of options, efficient algorithms need to be found. As almost all problems defined by Fluschnik and Kellerhals [FK21] are computationally hard, more precisely *NP-hard*, there is little hope to find general algorithms that reliably construct exact optimum solutions with reasonable runtime performance.

However, full generality is not required for deciding where to best place green bridges in real-world scenarios. For example, since specific geographic areas are (slightly curved) planes on the earth's surface, data on fragmented habitats and potential locations for green bridges can be expected to be *planar*.

In this thesis, we study input data restrictions for problems defined by Fluschnik and Kellerhals [FK21], focusing on problems that require habitats to be connected densely. Moreover, we define and study a new class of problems that can be seen as a combination of two problem classes introduced by Fluschnik and Kellerhals [FK21].

1.1 Problem Definition

We consider the following problem families.[1]

Problem: Π GREEN BRIDGES PLACEMENT WITH COSTS (Π GBP-C)

Input: An undirected graph G with edge costs $c : E(G) \to \mathbb{N}_0$, a set $\mathcal{H}$ of habitats where $H \subseteq V(G)$ for every habitat $H \in \mathcal{H}$, and an integer $k \in \mathbb{N}$.

Question: Is there an edge subset $F \subseteq E(G)$ with $\sum_{e \in F} c(e) \leq k$ such that for every habitat $H \in \mathcal{H}$ it holds that $H \subseteq V(G[F])$ and

$$\begin{aligned} \Pi \equiv d\text{-}\textsc{Reach}: &\quad G[F]^d[H] \text{ is connected?} \\ \Pi \equiv d\text{-}\textsc{Closed}: &\quad G[F]^d[H] \text{ is a clique?} \\ \Pi \equiv \ell\text{-}\textsc{Diam(eter)}: &\quad \operatorname{diam}(G[F][H]) \leq \ell? \\ \Pi \equiv (d,\ell)\text{-}\textsc{Closed}: &\quad \operatorname{diam}(G[F]^d[H]) \leq \ell? \end{aligned}$$

We refer to the respective unit cost versions by writing Π GBP instead of Π GBP-C. The problem family of (d, ℓ)-CLOSED GBP is newly defined here, whereas the other three problem families are taken from Fluschnik and Kellerhals [FK21]. We actively study 2-DIAM GBP and $(2, 2)$-CLOSED GBP in this thesis. Nevertheless, due to equivalences between the problems under some input restrictions, we also obtain results that can be directly transferred to 1-REACH GBP, 2-REACH GBP, and 2-CLOSED GBP.

The relationship of the abstract problems to the concrete issue of placing green bridges is as follows. Each vertex of the input graph G represents a patch of land bordered by barriers such as railroad tracks and motorways. Two vertices are adjacent if it is possible to build a green bridge connecting the corresponding patches. The individual problems can be characterized by the assumptions about animal movement behavior and connectivity requirements associated with them.

[1]The *d-th power* G^d of a graph G is a graph with the same vertex set as G and an edge set satisfying the property that two distinct vertices are adjacent in G^d if and only if their distance in G is at most d. Further definitions relating to graph theory can be found in Section 2.1.

For 2-Diam GBP the assumption about animal movement is that animals do not travel through patches that are not part of their habitat. The requirement on connectivity is that each animal must be able to move from any patch of its habitat to any other patch of its habitat by crossing at most two green bridges.

For $(2, 2)$-Closed GBP the assumption regarding the movement of animals is that animals move by making *hops*. A hop is a short journey in which an animal travels from one patch of its habitat to another patch of its habitat by visiting at most one patch in between. The patch visited in between can be a non-habitat patch. The connectivity requirement of $(2, 2)$-Closed GBP is that each animal must be able to travel from any patch of its habitat to any other patch of its habitat by making at most two hops.

1.2 Related Work

Herkenrath [Her21] and Herkenrath et al. [Her+22] study restrictions for 1-Reach GBP with a focus on restricted habitat structure. Other computational approaches to link multiple fragmented habitats have also been explored [Lai+11; LeB+13].

The need to "connect fragmented habitats" appears in a wide range of application areas, usually in the form of a special case of 1-Reach GBP. These areas include computer networks [Cho+07], social networks [AAR10], graph drawing [Bra+12], combinatorial auctions [CDS04], reconfigurable computing [Fan+08], vacuum technology [DK95], and structural biology [Aga+13], with Chockler et al. [Cho+07] calling for algorithms that connect habitats at small diameter. The wide range of occurrences has been pointed out by Chen et al. [Che+15]. Another problem concerned with connecting more than one habitat is Steiner Forest [BHM11], a generalization of the well-known Steiner Tree problem.

There are several related problems where the task is to connect only a single "habitat". Plesnik [Ple81] shows NP-hardness of connecting one habitat at minimum diameter given a cost budget. Schoone, Bodlaender, and Van Leeuwen [SBVL87] study adding no more than k edges to a graph such that the resulting graph does not have a diameter greater than a given integer ℓ. This problem is W[2]-hard with respect to the solution size k if $\ell = 2$ [Fra+15]. Gouveia [Gou98] studies a one-habitat version of $(2, 2)$-Closed GBP-C where the graph induced by the habitat must be a star with a given vertex as its center.

1.2.1 2-Diam GBP

Fluschnik and Kellerhals [FK21] study the parameterized complexity[2] of 2-Diam GBP with respect to the parameters solution size k and number of habitats r. Moreover, they consider the combined parameters $k + r$ and $\Delta + r$ where Δ is the maximum degree of the input graph G.

Theorem 1.1 ([FK21])**.** 2-Diam GBP *is fixed-parameter tractable with respect to k but admits no kernel of size polynomial in k unless* NP $\subseteq$ coNP/poly.

[2]Definitions relating to parameterized complexity can be found in Section 2.2.

Theorem 1.2 ([FK21])**.** 2-Diam GBP *is* NP*-hard even if* $r = 1$.

Theorem 1.3 ([FK21])**.** 2-Diam GBP *admits a kernel of size polynomial in* $k + r$.

Theorem 1.4 ([FK21])**.** 2-Diam GBP *admits a kernel of size polynomial in* $r + \Delta$.

Jansson, Levcopoulos, and Lingas [JLL21] give a polynomial-time approximation algorithm for 2-Diam GBP-C with an approximation factor of $\left(\binom{q}{2} - q + 2\right)\binom{q}{2}$ where $q = \max_{H \in \mathcal{H}} |H|$. Moreover, heuristic algorithms for 2-Diam GBP on cliques are studied in multiple papers [ÖLD17; OR11; OR16]. Besides finding a small solution F, the aim of these heuristic algorithms is for the maximum degree of $G[F]$ to be low.

Gionis et al. [Gio+17] consider a variant of 2-Diam GBP called SparseStars where an edge subset F is a solution if for each $H \in \mathcal{H}$ the graph $G[F][H]$ contains a spanning star. Note that this implies that $\mathrm{diam}(G[F][H]) \leq 2$ for every solution F and every habitat $H \in \mathcal{H}$. They give a polynomial-time approximation algorithm for SparseStars. Herrendorf [Her22] studies the parameterized complexity of SparseStars with respect to the parameters solution size $|F|$, number of edges removed $|E(G)| - |E(G[F])|$, and feedback edge number of $G[F]$. Korach and Stern [KS08] examine a variant of SparseStars which additionally requires that the solution F induces a spanning tree of G.

1.2.2 2-Diam GBP with Habitats of Size at Most Three

The following has been shown independently by multiple authors.

Theorem 1.5 ([Fan+08; Her+22; Hos+12])**.** 2-Diam GBP *is* NP*-hard even if each habitat has size at most three and* G *is a clique.*

Herkenrath et al. [Her+22] and Korach and Stern [KS03] identify special cases that can be solved in polynomial time.

Theorem 1.6 ([Her+22])**.** 2-Diam GBP-C *can be solved in polynomial time if* G *is plane and each habitat induces a triangle which is the boundary of a face.*

Theorem 1.7 ([Her+22])**.** 2-Diam GBP-C *can be solved in polynomial time on graphs of maximum degree three if each habitat induces a triangle.*

Theorem 1.8 ([KS03])**.** 2-Diam GBP-C *can be solved in linear time if each habitat has size at most three and there exists a solution* F *that induces a spanning tree of* G.

Hosoda et al. [Hos+12] examine approximability. They show that the optimization version of 2-Diam GBP[3] with habitats of size at most three and G being a clique is APX-complete and that it admits no polynomial-time $(2 - \epsilon)$-approximation algorithm for any $\epsilon > 0$ unless the unique games conjecture fails. However, they also show that a polynomial-time 2-approximation algorithm exists.

[3]The task of the optimization version is to find an edge subset $F \subseteq E(G)$ of minimum size such that for every habitat $H \in \mathcal{H}$ it holds that $H \subseteq V(G[F])$ and $\mathrm{diam}(G[F][H]) \leq 2$.

1.2.3 (2, 2)-Closed GBP with Habitats of Size at Most Two

To the best of our knowledge, $(2,2)$-Closed GBP-C has not been studied before. However, if every habitat has size two, then $(2,2)$-Closed GBP-C coincides with the 2-Path Network problem introduced by Dahl and Johannessen [DJ04].

Theorem 1.9 ([DJ04])**.** $(2,2)$-Closed GBP-C *is* NP*-hard even if G is a clique and every habitat has size two.*

Theorem 1.10 ([DJ04])**.** $(2,2)$-Closed GBP-C *can be solved in polynomial time if every habitat has size two and habitats are pairwise disjoint.*

Various authors study heuristic algorithms for $(2,2)$-Closed GBP-C with habitats of size two [Bar+13; Clí+19; DJ04; RR02].

1.3 Our Contributions and the Structure of This Thesis

Like the previous work done by Fluschnik and Kellerhals [FK21] and Herkenrath et al. [Her+22], we exclusively study exact algorithms under structural restrictions or within the framework of parameterized algorithms.

The research part of this work is divided into two parts, with the first (Chapter 3) being on 2-Diam GBP and the second (Chapter 4) being on $(2,2)$-Closed GBP. Each of these chapters opens with a section providing general reduction rules for the corresponding problem (Sections 3.1 and 4.1). When we consider hardness, we almost always do so for the respective unit cost versions, whereas we provide algorithms for the edge-weighted versions. For 2-Diam GBP we study two restrictions in particular depth.

The first of these two restrictions is bounded maximum degree (Section 3.2). We give a linear-time algorithm for 2-Diam GBP-C on graphs of maximum degree at most three. For maximum degree five and above, we show that 2-Diam GBP is NP-hard even if each habitat has size at most three. This leaves the case where input graphs have maximum degree at most four as a gap. We close this gap for the more restricted case where in addition to the input graph having maximum degree at most four it holds that each habitat has size at most three, which can be solved in polynomial time. Thus, for 2-Diam GBP with habitats of size at most three we obtain a full dichotomy regarding maximum degree, thereby answering an open question from Herkenrath et al. [Her+22].

The second restriction we closely examine is bounded maximum habitat size combined with planarity of input graphs (Section 3.3). We prove that 2-Diam GBP-C on planar graphs can be solved in polynomial time if each habitat has size at most three. This resolves another question from Herkenrath et al. [Her+22]. On the other hand, we show that 2-Diam GBP on planar graphs is NP-hard even if each habitat has size at most four (and the maximum degree of the input graph is at most five). These results constitute a further dichotomy.

Subsequently, we briefly explore additional structural parameterizations of 2-Diam GBP (Section 3.4), including the feedback edge number and vertex cover number.

For $(2,2)$-Closed GBP we do not focus on particular complexity dichotomies as much as for 2-Diam GBP. Because of this, the chapter on $(2,2)$-Closed GBP is more conventionally divided in a section on tractability (Section 4.2) and a section on intractability (Section 4.3). Among other findings, we show that $(2,2)$-Closed GBP is NP-hard even on planar graphs of maximum degree at most four with each habitat having size at most two.

We summarize our findings in Figures 1.1 and 1.2.

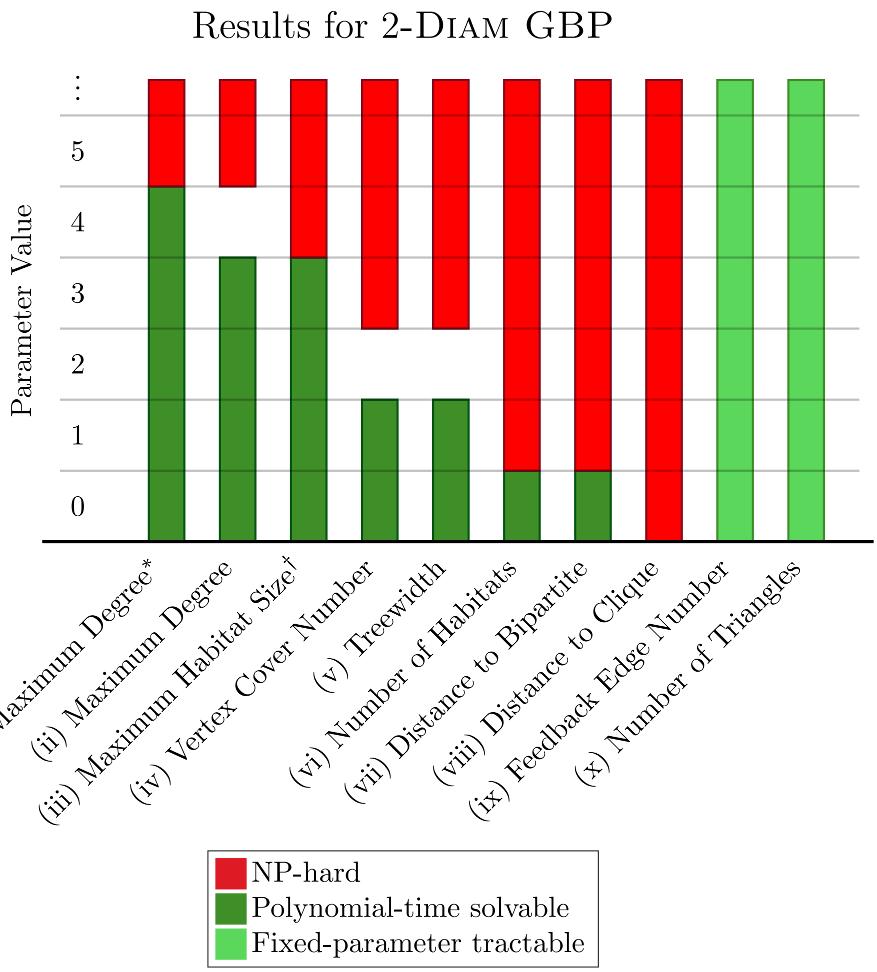


Figure 1.1: Overview of our results regarding 2-DIAM GBP. For each of the parameters shown on the bottom side, we represent its computational complexity with respect to the parameter values shown on the left side using red and green bars. The depicted hardness-results (red) refer to the problem version without edge costs, whereas the tractability-results (shades of green) refer to the problem version with edge costs. In the following, p. and h. stand for "polynomial-time solvability" and "NP-hardness", respectively. (i) p.: [Sec. 3.2.2]; h.: [Sec. 3.2.3]. (ii) p.: [Sec. 3.2.1]; h.: [Sec. 3.2.3]. (iii) p.: [Sec. 3.3.1]; h.: NP-hard even on graphs of maximum degree five [Sec. 3.3.2]. (iv) p.: [Sec. 3.4.1]; h.: [Sec. 3.4.4]. (v) p.: [Sec. 3.4.1]; h.: [Sec. 3.4.4]. (vi) p.: trivial; h.: [FK21] (vii) p.: [Sec. 3.4.1]; h.: [Sec. 3.4.4]. (viii) h.: NP-hard even if each habitat has size at most three [Her+22], [Sec. 3.4.3]. (ix) [Sec. 3.4.2], but there exists no polynomial kernel unless NP $\subseteq$ coNP/poly [Prop. 3.84]. (x) [Sec. 3.4.1], but there exists no polynomial kernel unless NP $\subseteq$ coNP/poly [Prop. 3.84].
*(If each habitat has size at most three) †(On planar graphs)

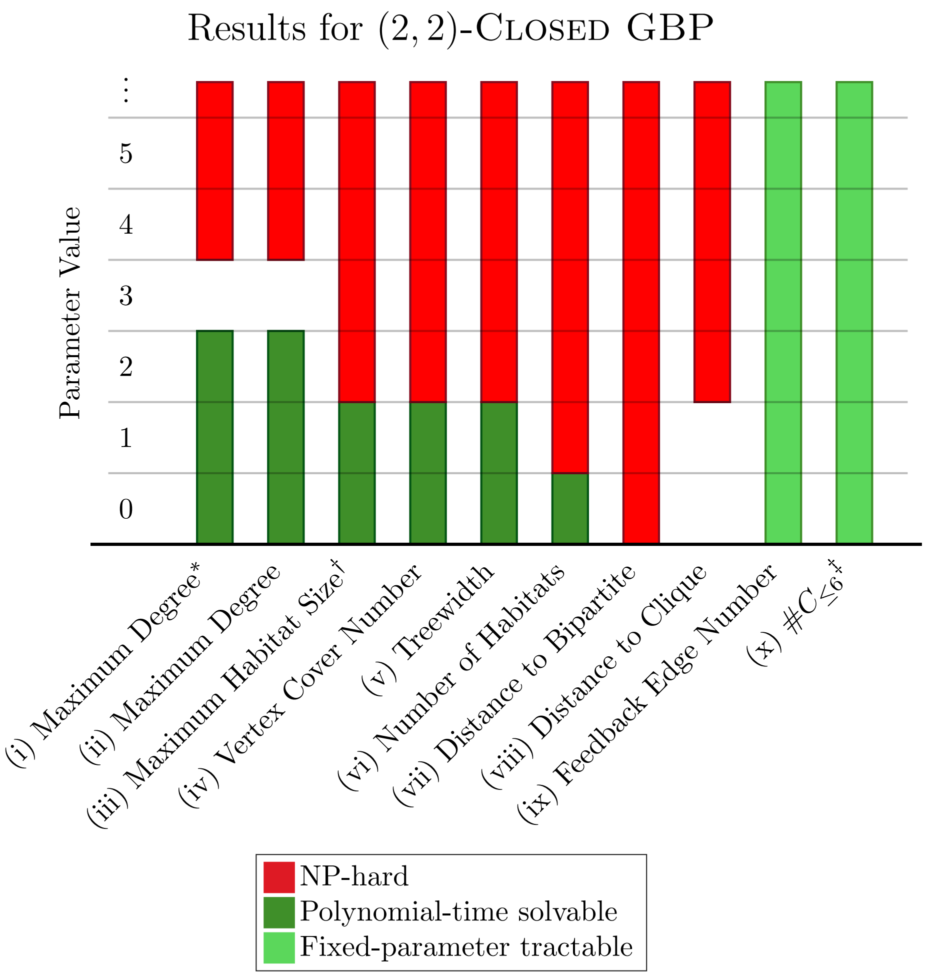


Figure 1.2: Overview of our results regarding $(2,2)$-Closed GBP. For each of the parameters shown on the bottom side, we represent its computational complexity with respect to the parameter values shown on the left side using red and green bars. The depicted hardness-results (red) refer to the problem version without edge costs, whereas the tractability-results (shades of green) refer to the problem version with edge costs. In the following, p. and h. stand for "polynomial-time solvability" and "NP-hardness", respectively. (i) p.: [Sec. 4.2.1]; h.: [Sec. 4.3.4]. (ii) p.: [Sec. 4.2.1]; h.: [Sec. 4.3.4]. (iii) p.: trivial; h.: [Sec. 4.3.4]. (iv) p.: [Sec. 4.2.1]; h.: [Sec. 4.3.1]. (v) p.: [Sec. 4.2.1]; h.: [Sec. 4.3.1]. (vi) p.: trivial; h.: [Sec. 4.3.3] (vii) h.: [Sec. 4.3.3]. (viii) See the note§ below. (ix) [Sec. 4.2.2], but there exists no polynomial kernel unless NP $\subseteq$ coNP/poly [Prop. 4.16]. (x) [Sec. 4.2.2], but there exists no polynomial kernel unless NP $\subseteq$ coNP/poly [Prop. 4.16].
§(In this thesis, we prove that $(2,2)$-Closed GBP is NP-hard on graphs with distance to clique two [Sec. 4.3.2]. Dahl and Johannessen [DJ04] show that the version with edge costs is NP-hard even on cliques. As a side note, we mention that the version without edge costs, however, can be solved in polynomial time on cliques. The reason for this is that the edge set of any spanning star of $G[\bigcup_{H\in\mathcal{H}} H]$ is a solution.)
*(If each habitat has size at most two) †(On planar graphs) ‡(The number of cycles of length at most six is denoted by $\#C_{\leq 6}$.)

Chapter 2

Preliminaries

In this chapter, we introduce basic notation and definitions. For an extensive introduction to graph theory, see, e.g., Diestel [Die17]. For more information on parameterized complexity, refer to the standard textbooks [Cyg+15; DF13; FG06; Nie06].

Sets. We use $\mathbb{N} \coloneqq \{1, 2, 3, \dots\}$ to denote the natural numbers without zero and $\mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\}$. For sets $X_1, \dots, X_n, Y$ we write $X_1 \uplus \dots \uplus X_n = Y$ if $X_1 \cup \dots \cup X_n = Y$ and $X_1 \cap \dots \cap X_n = \emptyset$. For sets X, Y we define the *symmetric difference* $X \triangle Y \coloneqq (X \setminus Y) \cup (Y \setminus X)$. For a set X, we denote the set of two-element subsets of X by $[X]^2$.

2.1 Graph Theory

A *(finite) graph* is a pair $G = (V, E)$ with a finite *vertex set* V and *edge set* $E \subseteq [V]^2$. We also denote the vertex set of a graph G by $V(G)$ and the edge set by $E(G)$. For an edge $e \in E(G)$, we call the vertices in e the *endvertices* of e.

Subgraphs. Let G be a graph. A *subgraph* H of G is a graph with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. To express that H is a subgraph of G, we write $H \subseteq G$. If H is a subgraph of G, then we say that G *contains* H. Given a vertex subset $U \subseteq V(G)$, the *induced subgraph* $G[U]$ of G on U is the graph with vertex set U and edge set $\{e \in E(G) \mid e \subseteq U\}$. We say that U *induces* $G[U]$. Moreover, we define $G - U \coloneqq G[V(G) \setminus U]$. Given an edge subset $F \subseteq E$, the *induced subgraph* $G[F]$ of G on F is the graph with vertex set $\{v \in V(G) \mid \exists e \in F.\, v \in e\}$ and edge set F. We say that F *induces* $G[F]$. Moreover, we define $G - F \coloneqq (V(G), E(G) \setminus F)$.

Intersection and Union of Graphs. The *intersection* $G \cap H$ of two graphs G and H is the graph with vertex set $V(G) \cap V(H)$ and edge set $E(G) \cap E(H)$. The *union* $G \cup H$ of two graphs G and H is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$.

Paths and Cycles. A *path* P is a graph with vertex set $\{v_1, \dots, v_n\}$ and edge set $\{\{v_i, v_{i+1}\} \mid i \in \{1, \dots, n-1\}\}$ where $n \geq 2$. The vertices v_1 and v_n are the *endvertices* of P. We call a path with endvertices v_1 and v_n a (v_1, v_n)-*path* and denote it by P_n. The *length* of a path is the number of edges it has. (Note that the length of a P_n is $n-1$ and not n.) We obtain a *cycle* by adding the edge $\{v_1, v_n\}$ to a (v_1, v_n)-path with $n \geq 3$ and denote it by C_n. The *length* of a cycle is the number edges it has. A cycle of length three is called a *triangle*. If a graph C is a cycle, then we call an element $e \in [V(C)]^2$ with $e \notin E(C)$ a *chord* of C. If a cycle C is a subgraph of a graph G and $E(G)$ contains no chord of C, then C is *chordless* in G.

Connectedness. Let G be a graph. We say that G is *connected* if G contains a (u, v)-path for any two vertices $u, v \in V(G)$. A connected subgraph $C \subseteq G$ is a *component* of G if no graph $C' \subseteq G$ with $C \neq C'$ and $C \subseteq C'$ exists. A vertex subset $X \subseteq V(G)$ *separates* the vertex subsets $A \subseteq V(G)$ and $B \subseteq V(G)$ in G if for every vertex $a \in A$ and every vertex $b \in B$ it holds that there is no component $C \subseteq G - X$ with $a, b \in V(C)$. A vertex subset $X \subseteq V(G)$ *separates* G if there are two vertices $a, b \in V(G) \setminus X$ such that X separates $\{a\}$ and $\{b\}$. The graph G is *k-connected* for a $k \in \mathbb{N}$ if there is no vertex subset $X \subseteq V(G)$ with $|X| \leq k$ such that X separates G.

Distance and Diameter. Let G be a graph. The *distance* $\operatorname{dist}_G(u, v)$ between vertices $u, v \in V(G)$ in G is the length of a shortest (u, v)-path in G. We write $\operatorname{dist}_G(u, v) = 0$ if $u = v$ holds. We write $\operatorname{dist}_G(u, v) = \infty$ if there is no (u, v)-path in G. The *diameter* of G, denoted by $\operatorname{diam}(G)$, is the largest distance between any two vertices of G.

Vertex Degree and Neighborhoods. Let G be a graph. An edge $e \in E(G)$ is *incident* to a vertex $v \in V(G)$ if v is an endvertex of e. We say that two vertices $u, v \in V(G)$ are *adjacent* or *neighbors* if $\{u, v\} \in E(G)$. The *degree* $\deg_G(v)$ of a vertex $v \in V(G)$ in G is the number of vertices adjacent to v. A vertex $v \in V(G)$ with $\deg_G(v) = 0$ is *isolated* in G. We say that a graph G is *cubic* if for every $v \in V(G)$ it holds that $\deg_G(v) = 3$. The *maximum degree* $\Delta(G)$ of graph G is defined as $\Delta(G) \coloneqq \max\{\deg_G(v) \mid v \in V(G)\}$. The *open neighborhood* $N_G(v)$ of a vertex $v \in V(G)$ in G is the set of vertices adjacent to v. The *closed neighborhood* $N_G[v]$ of a vertex $v \in V(G)$ in G is defined as $N_G[v] \coloneqq N_G(v) \cup \{v\}$.

Power Graph and Wide Neighborhoods. The *k-th power* G^k of graph G is the graph with vertex set $V(G)$ and edge set $\{\{u, v\} \in [V(G)]^2 \mid \operatorname{dist}_G(u, v) \leq k\}$. The *open k-neighborhood* $N_G^k(v)$ of a vertex $v \in V(G)$ in G is defined as $N_G^k(v) \coloneqq N_{G^k}(v)$. The *closed k-neighborhood* $N_G^k[v]$ of a vertex $v \in V(G)$ in G is defined as $N_G^k[v] \coloneqq N_G^k(v) \cup \{v\}$. The *$k$-neighborhood* of a vertex subset $U \subseteq V(G)$ in G is defined as $N_G^k[U] \coloneqq \bigcup_{u \in U} N_G^k[u]$.

Forests, Trees, Stars, and Claws. A *forest* F is a graph that contains at most one (u, v)-path for any two vertices $u, v \in V(F)$. A *tree* is a connected forest. A *star* is a tree of maximum diameter at most two. If a vertex of a star has degree larger than one, then it is the *center vertex* of the star. A *claw* is a star with three edges. A graph is *claw-free* if it does not contain a claw as an induced subgraph.

Empty Graph, Clique, Bipartite Graph. The *empty graph* is the graph with an empty vertex set. A *clique* K is a graph with edge set $[V(K)]^2$. We call a graph B *bipartite* if there are sets X, Y with $X \uplus Y = V(B)$ such that $E(B) \subseteq \{\{x, y\} \mid x \in X \wedge y \in Y\}$.

Series-Parallel Graphs. A *series-parallel* graph G is a graph with two distinguished vertices denoted by x_G and y_G obtained by the following rules.

- A path P with a vertex set of size two is series-parallel and the two distinguished vertices x_P and y_P are the two vertices of P.
- Let G and H be vertex-disjoint series-parallel graphs. If we set $x_G \coloneqq x_H$ and $y_G \coloneqq y_H$, then the graph $I \coloneqq G \cup H$ is series-parallel with $x_I \coloneqq x_G = x_H$ and $y_I \coloneqq y_G = y_H$. (This is called *parallel composition.*)
- Let G and H be vertex-disjoint series-parallel graphs. If we set $y_G \coloneqq x_H$, then the graph $I \coloneqq G \cup H$ is series-parallel with $x_I \coloneqq x_G$ and $y_I \coloneqq y_H$. (This is called *series composition.*)

Weighted Graphs. A *vertex-weighted graph* is a pair (G, c) where G is a graph and $c : V(G) \to \mathbb{Z}$ is a function assigning a cost to every vertex in $V(G)$. An *edge-weighted graph* is a pair (G, c) where G is a graph and $c : E(G) \to \mathbb{Z}$ is a function assigning a cost to every edge in $E(G)$. Given an edge subset $F \subseteq E(G)$, we use the notation $c(F) \coloneqq \sum_{e \in F} c(e)$.

Graph Isomorphism. A graph G is *isomorphic* to a graph H if there is a bijective function $f : V(G) \to V(H)$ such that for each pair of vertices $u, v \in V(G)$ it holds that $\{u, v\} \in E(G)$ if and only if $\{f(u), f(v)\} \in E(H)$.

Planarity. Let G be a graph. Let π be a function that assigns a unique point $\pi(v) \in \mathbb{R}^2$ to every vertex $v \in V(G)$ and an arc $\pi(\{u, v\}) \subseteq \mathbb{R}^2$ to every edge $\{u, v\} \in E(G)$ such that the arc $\pi(\{u, v\})$ has the endpoints $\pi(u)$ and $\pi(v)$ and does not contain any point $\pi(w)$ with $w \in V(G) \setminus \{u, v\}$. If no point $p \in \mathbb{R}^2$ exists such that p is an interior point of two arcs $\pi(e)$ and $\pi(e')$ with distinct $e, e' \in E(G)$, then π is a *planar embedding* of G. We call G *planar* if a planar embedding of G exists. Given a planar embedding π of G, the pair $D \coloneqq (U, F)$ with $U \coloneqq \{\pi(v) \mid v \in V(G)\}$ and $F \coloneqq \{\pi(e) \mid e \in E(G)\}$ is a *drawing* of G. Although a drawing D of G is not formally a graph, we refer to the elements of U as vertices, to the elements of F as edges, and generally do not strictly distinguish between G and its drawing D. A *plane graph* is a drawing of some graph. If G is plane, then the contiguous areas of $\mathbb{R}^2 \setminus (V(G) \cup \bigcup E(G))$ are the *faces* of G.[1] A face f is an *inner face* if a disk $d \subseteq \mathbb{R}^2$ exists such that f is contained in d. If a face f is not an inner face, then we call it an *outer face*. If G is plane, then a cycle $C \subseteq G$ is the boundary of a face f (regarding G) if f is both a face of G and one of the contiguous areas of $\mathbb{R}^2 \setminus (V(C) \cup \bigcup E(C))$.

[1] The notation $\bigcup X$ for a collection X of sets is only used inside this paragraph for better readability. It is defined as $\bigcup \mathcal{X} \coloneqq \bigcup_{X \in \mathcal{X}} X$ and gives a "flattened" version of $\mathcal{X}$.

Treewidth. A *tree decomposition* of a graph G is a pair $(T, \mathcal{X})$ where T is a tree and $\mathcal{X} = (X_t)_{t \in V(T)}$ is a family of vertex sets $X_t \subseteq V(G)$ such that

- $V(G) = \bigcup_{t \in V(T)} X_t$,
- for each edge $e \in E(G)$ there is a vertex $t \in V(T)$ with $e \subseteq X_t$, and
- for each vertex $v \in V(G)$ it holds that the induced graph $T[\{t \in V(T) \mid v \in X_t\}]$ is a tree.

The members of $\mathcal{X}$ are called *bags*. The *width* of a tree decomposition $(T, \mathcal{X})$ is the number $\max\{|X_t| - 1 \mid t \in V(T)\}$. The *treewidth* of a graph G is the smallest width of all tree decompositions of G.

Independent Set, Vertex Cover, and Feedback Edge Set. Let G be a graph. An *independent set* of G is a vertex subset $S \subseteq V$ such that for every edge $\{u, v\} \in E(G)$ it holds that $u \notin S$ or $v \notin S$. A *vertex cover* of G is a vertex subset $S \subseteq V$ such that for every edge $\{u, v\} \in E(G)$ it holds that $u \in S$ or $v \in S$. A *feedback edge set* of G is an edge subset $F \subseteq E(G)$ such that $G - F$ is a forest.

Some Graph Parameters. Let G be a graph. The *vertex cover number* of G is the size of a minimum size vertex cover of G. The *feedback edge number* of G is the size of a minimum size feedback edge set of G. The *distance to clique* of G is the minimum number of vertices that need to be deleted for G to become a clique. The *distance to bipartite* of G is the minimum number of vertices that need to be deleted for G to become bipartite.

2.2 Parameterized Complexity

Let Σ be a finite alphabet. A *parameterized problem* Π is a set $\Pi \subseteq \Sigma^* \times \mathbb{N}$ of pairs. An element $(x, k) \in \Sigma^* \times \mathbb{N}$ is called an *instance* of Π. The number k is the *parameter* of the instance (x, k). If there is a computable function $f : \mathbb{N} \to \mathbb{N}$ and an algorithm $\mathcal{A}$ that decides for any instance (x, k) of Π whether $(x, k) \in \Pi$ using $f(k) \cdot |x|^{\mathcal{O}(1)}$ time, then Π is *fixed-parameter tractable*.

Reduction Rule. Instances (x, k) and (x', k') of Π are *equivalent* if $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$. A *reduction rule* $\mathcal{R}$ for problem Π is a polynomial-time algorithm that receives an instance (x, k) of Π as input and produces an instance (x', k') of Π as output. We say that reduction rule $\mathcal{R}$ is *correct* if (x, k) and (x', k') are equivalent.

Kernel. A *(problem) kernel* $\mathcal{K}$ for problem Π is a polynomial-time algorithm that receives an instance (x, k) of Π as input and produces an instance (x', k') as output such that $|x'| \leq g(k)$ and $k' \leq g(k)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$. We call g the *size* of $\mathcal{K}$. If g is polynomial, then $\mathcal{K}$ is a *polynomial kernel*.

Polynomial Parameter Transformation. Let Π and Π' be two parameterized problems. An instance (x, k) of Π and an instance (x', k') of Π' are *equivalent* if $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi'$. A *polynomial parameter transformation* from Π to Π' is a polynomial-time algorithm that receives an instance (x, k) of Π as input and produces an equivalent instance (x', k') of Π' as output such that $k' \leq p(k)$ for some polynomial function $p : \mathbb{N} \to \mathbb{N}$.

Existence of Polynomial Kernels. Let Π, Π' be two parameterized problems with the property that the unparameterized versions of Π and Π' are NP-complete. If there is a polynomial parameter transformation from Π to Π' and Π' admits a polynomial kernel, then Π also admits a polynomial kernel [BTY11]. We use this fact later to show the nonexistence of polynomial kernels for some parameterized problems under the complexity-theoretic assumption that NP $\subseteq$ coNP/poly is not true, which is considered very likely.

Chapter 3

2-Diam GBP

3.1 Preprocessing

In this section, we define and analyze reduction rules for later usage. We consider an instance $\mathcal{I} = (G, c, \mathcal{H}, k)$ of 2-Diam GBP-C. Let $n := |V(G)|$ and let $m := |E(G)|$. Moreover, let $q := \max_{H \in \mathcal{H}} |H|$ be the size of the largest habitat. Whenever a reduction rule modifies the edge set $E(G)$ and there is no specification given on how the cost function c changes, we assume that c is adjusted in the most immediate way.

Reduction Rule 3.1. *If for a vertex $v \in V(G)$ there is no habitat $H \in \mathcal{H}$ with $v \in H$, then delete v.*

Observation 3.2. *Reduction Rule 3.1 is correct and can be applied exhaustively in $\mathcal{O}(n + m + rq)$ time.*

Proof. Let $\mathcal{I}'$ be the instance of 2-Diam GBP-C obtained from $\mathcal{I}$ by application of Reduction Rule 3.3. Then, a solution to $\mathcal{I}$ is a solution to $\mathcal{I}'$ and vice versa.

To achieve the stated running time, we do the following. For every habitat $H \in \mathcal{H}$ and every vertex $v \in H$, we mark the vertex v. This takes $\mathcal{O}(rq)$ time. Then, we iterate through all vertices $v \in V(G)$ deleting every unmarked vertex. This takes $\mathcal{O}(n)$ time. ☐

Reduction Rule 3.3. *If for an edge $e \in E(G)$ there is no habitat $H \in \mathcal{H}$ with $e \subseteq H$, then delete e.*

Observation 3.4. *Reduction Rule 3.3 is correct and can be applied exhaustively in $\mathcal{O}(n + m + rq^2)$ time.*

Proof. Let $\mathcal{I}' = (G', c', \mathcal{H}', k')$ be the instance of 2-Diam GBP-C obtained from $\mathcal{I}$ by application of Reduction Rule 3.3. We show that $\mathcal{I}$ is a `yes`-instance if and only if $\mathcal{I}'$ is a `yes`-instance. Let $e \in E(G)$ be the edge deleted in the application of Reduction Rule 3.3.

($\Rightarrow$) Let F be a solution to $\mathcal{I}$. We claim that $F' := F \setminus \{e\}$ is a solution to $\mathcal{I}'$. Clearly, $c(F') \leq k'$. Assume towards a contradiction that there is a habitat $H \in \mathcal{H}$ with vertices $u, v \in H$ such that $\mathrm{dist}_{G[F'][H]}(u, v) > 2$. Since $\mathrm{dist}_{G[F][H]}(u, v) \leq 2$, this implies that $e \subseteq H$, a contradiction.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. Then, F is also a solution to $\mathcal{I}$.

To achieve the stated running time, we do the following. For every habitat $H \in \mathcal{H}$ and every pair of vertices $u, v \in H$ with $\{u, v\} \in E(G)$, we mark the edge $\{u, v\}$. This takes $\mathcal{O}(rq^2)$ time. Then, we iterate through all edges $e \in E(G)$ deleting every unmarked edge. This takes $\mathcal{O}(m)$ time. □

Reduction Rule 3.5. *If a habitat $e \in \mathcal{H}$ induces a P_2, then first set $k := k - c(e)$, after that set $c(e) := 0$, and finally delete the habitat e from $\mathcal{H}$.*

Observation 3.6. *Reduction Rule 3.5 is correct and can be applied exhaustively in $\mathcal{O}(n + m + r)$ time.*

Proof. Let $\mathcal{I}' = (G', c', \mathcal{H}', k')$ be the instance of 2-Diam GBP-C obtained from $\mathcal{I}$ by application of Reduction Rule 3.5. We show that $\mathcal{I}$ is a `yes`-instance if and only if $\mathcal{I}'$ is a `yes`-instance. Let $e \in \mathcal{H}$ be the habitat deleted in the application of Reduction Rule 3.5.

($\Rightarrow$) Let F be a solution to $\mathcal{I}$. We claim that F is also a solution to $\mathcal{I}'$. Clearly, it holds that $\operatorname{diam}(G'[F][H]) \leq 2$ for every $H \in \mathcal{H}'$. Thus, we only need to show that $c'(F) \leq k'$. Because of $\operatorname{diam}(G[F][e]) \leq 2$, it holds that $e \in F$. Hence,

$$c'(F) = c'(F \setminus \{e\}) = c(F \setminus \{e\}) = c(F) - c(e) \leq k'.$$

($\Leftarrow$) Let F' be a solution to $\mathcal{I}'$. We claim that $F := F' \cup \{e\}$ a solution to $\mathcal{I}$. Clearly, it holds that $\operatorname{diam}(G[F][H]) \leq 2$ for every $H \in \mathcal{H}$. It is left to show that $c(F) \leq k$. It holds that

$$\begin{aligned} c(F) &= c(F \setminus \{e\}) + c(e) \\ &= c'(F \setminus \{e\}) + c(e) \\ &= c'(F) + c(e) \\ &\leq k' + c(e) \\ &= k. \end{aligned}$$

Since the most time-consuming action is to iterate through all habitats, the linear running time is immediate. □

Reduction Rule 3.7. *If for a habitat $H \in \mathcal{H}$ it holds that $G[H]$ is a P_3, then delete H from $\mathcal{H}$ and extend H by the edge set of $G[H]$.*

Observation 3.8. *Reduction Rule 3.7 is correct and can be applied exhaustively in $\mathcal{O}(n + m + r)$ time.*

Proof. Let $\mathcal{I}' = (G', c', \mathcal{H}', k')$ be the instance of 2-Diam GBP-C obtained from $\mathcal{I}$ by application of Reduction Rule 3.7. We show that $\mathcal{I}$ is a `yes`-instance if and only if $\mathcal{I}'$ is a `yes`-instance. Let $H \in \mathcal{H}$ be the habitat deleted in the application of Reduction Rule 3.7.

($\Rightarrow$) Let F be a solution to $\mathcal{I}$. We claim that F is also a solution to $\mathcal{I}'$. Assume towards a contradiction that F is not a solution to $\mathcal{I}'$. Then, for one of the edges $e \in E(G[H])$ it holds that $e \notin F$. This implies that $\operatorname{diam}(G[F][H]) = \infty$, a contradiction to F being a solution to $\mathcal{I}$.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. We claim that F is also a solution to $\mathcal{I}$. Assume towards a contradiction that F is not a solution to $\mathcal{I}$. Then, for one of $e \in E(G[H])$ it holds that $e \notin F$. This implies that $\operatorname{diam}(G[F][e]) = \infty$, a contradiction to F being a solution to $\mathcal{I}'$

Since the most time-consuming action is to iterate through all habitats, the linear running time is immediate. $\square$

Most algorithms in this thesis are based on constructing a solution F for the input instance $\mathcal{I}$. In this context, it is desirable to define procedures that can add edges to the solution F that is being constructed. Such a procedure is not technically a reduction rule since it slightly changes the problem 2-Diam GBP-C by adding the condition that certain edges need to be included in a solution. To keep things more uniform and simple, we nevertheless refer to a procedure that fixes an edge to be included in the solution F as a reduction rule.

Reduction Rule 3.9. *If for an edge $e \in E(G)$ there is a habitat $H \in \mathcal{H}$ with $e \subseteq H$ such that e is not contained in a triangle in G, then fix $e \in F$.*

Observation 3.10. *Reduction Rule 3.9 is correct and can be applied exhaustively in $\mathcal{O}(n + m + rq^3)$ time.*

Proof. Let $\{u, v\} = e \in E(G)$ be the edge fixed to be included in F in the application of Reduction Rule 3.9. Assume towards a contradiction that there is a solution $\widetilde{F}$ to $\mathcal{I}$ with $e \notin \widetilde{F}$. By definition of 2-Diam GBP-C, there is a (u, v)-path of length at most two in $G[\widetilde{F}]$. Since $e \notin \widetilde{F}$, this implies that there is a vertex $w \in V(G)$ such that $\{u, w\}, \{w, v\} \in E(G)$. Therefore, $G[\{u, v, w\}]$ is a triangle in G, a contradiction to the edge e not being contained in a triangle in G.

To achieve the stated running time, we do the following. For every habitat $H \in \mathcal{H}$ and every pair of vertices $u, v \in H$ with $\{u, v\} \in E(G)$, perform the following actions. Check whether there is a third vertex $w \in H$ such that $G[\{u, v, w\}]$ is a triangle. If no such vertex w exists, then fix $\{u, v\} \in F$. $\square$

The following reduction rule deletes components of small size in polynomial time. Because of how we will use this reduction rule, we are interested in eliminating components having at most six vertices.

Reduction Rule 3.11. *Let $C \subseteq G$ be a component of G with $|V(C)| \leq 6$. If there is a habitat $H \in \mathcal{H}$ with vertices $u, v \in H$ such that $u \in V(C)$ and $v \in V(G) \setminus V(C)$, then return a trivial* `no`*-instance. Let $\mathcal{H}_C \subseteq \mathcal{H}$ be the set of all habitats $H \in \mathcal{H}$ with $H \subseteq V(C)$. Search for a minimum cost local solution $F_C \subseteq E(C)$ satisfying $\operatorname{diam}(C[F_C][H]) \leq 2$ for every $H \in \mathcal{H}_C$. If no local solution has been found during the search, then return a trivial* `no`*-instance. Otherwise, delete C from G, delete every habitat in $\mathcal{H}_C$ from $\mathcal{H}$, and set $k := k - c(F_C)$.*

Observation 3.12. *Reduction Rule 3.11 is correct and can be applied exhaustively in $\mathcal{O}(n + m + rq)$ time.*

Proof. Let C be the component of G with $|V(C)| \leq 6$ selected in the application of Reduction Rule 3.11. If there is a habitat $H \in \mathcal{H}$ with vertices $u, v \in H$ such that $u \in V(C)$ and $v \in V(G) \setminus V(C)$, then it holds that $\mathrm{diam}(H) = \infty$ implying that $\mathcal{I}$ is a `no`-instance.

If no local solution exists, then there is a habitat $H \in \mathcal{H}$ such that $\mathrm{diam}(C[H]) > 2$. From $\mathrm{diam}(C[H]) > 2$ it follows that $\mathrm{diam}(G[H]) > 2$ and hence $\mathcal{I}$ is a `no`-instance.

In the case where a minimum cost local solution F_C was found, let $\mathcal{I}' := (G', \mathcal{H}', c', k')$ be the instance of 2-Diam GBP-C obtained from $\mathcal{I}$. We show that $\mathcal{I}$ is a `yes`-instance if and only if $\mathcal{I}'$ is a `yes`-instance. ($\Rightarrow$) Let F be a solution to $\mathcal{I}$. Then, $F \setminus E(C)$ is a solution to $\mathcal{I}'$. ($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. Then, $F \cup F_C$ is a solution to $\mathcal{I}$.

To achieve the stated running time, we can use the following approach. Using depth-first search we can create a table T_1 mapping each vertex $v \in V(G)$ to the component $C \subseteq G$ with $v \in V(C)$ in $\mathcal{O}(n + m)$ time. Using table T_1 we can determine for each habitat $H \in \mathcal{H}$ whether there are vertices $u, v \in H$ and a component $C \subseteq G$ such that $u \in V(C)$ and $v \in V(G) \setminus V(C)$ in overall $\mathcal{O}(rq)$ time. Moreover, using table T_1 we can create another table T_2 mapping each component $C \subseteq G$ to the set containing all habitats $H \in \mathcal{H}$ with $H \subseteq V(C)$ in $\mathcal{O}(rq)$ time. Clearly, the number of components $C \subseteq G$ is in $\mathcal{O}(n)$. Since we only consider components having at most six vertices, it follows that all potential local solutions can be constructed in $\mathcal{O}(n)$ time. As for each habitat $H \in \mathcal{H}$ there is only one component $C \subseteq G$ with $H \subseteq V(C)$, checking which potential local solutions are valid takes $\mathcal{O}(r)$ time. (We use table T_2 to quickly find the habitats corresponding to each component when doing these checks.) Taken together, the time needed for exhaustive application of Reduction Rule 3.11 is $\mathcal{O}(n+m+rq)$. □

3.2 On Graphs of Low Maximum Degree Δ

3.2.1 A Polynomial-Time Algorithm for $\Delta \leq 3$

We show that 2-Diam GBP-C on graphs of maximum degree at most three is solvable in linear time. We consider an instance $\mathcal{I} = (G, \mathcal{H}, c, k)$ of 2-Diam GBP-C.

Proposition 3.13. 2-Diam GBP-C *on graphs of maximum degree at most three can be solved in* $\mathcal{O}(|V(G)| + |E(G)| + |\mathcal{H}|)$ *time.*

We take a constructive approach, i.e., we try to find a solution F of minimum cost. We use multiple reduction rules. These reduction rules simplify the instance to a point where a minimum cost solution of the entire instance can be constructed by combining local minimum cost solutions. We then use this property to obtain F.

We describe the application of the first three reduction rules before giving some more detail on the general idea that we will follow in the further course of this proof. First, we use a reduction rule that helps us to apply some of the succeeding reduction rules in linear time.

Reduction Rule 3.14. *If a habitat* $H \in \mathcal{H}$ *with* $|H| > 10$ *exists, then return a trivial* `no`*-instance.*

Observation 3.15. *Reduction Rule 3.14 is correct.*

Proof. Assume towards a contradiction that $\mathcal{I}$ is a `yes`-instance and there is a habitat $H \in \mathcal{H}$ with $|H| > 10$. Then, it holds that $\operatorname{diam}(G[H]) \leq 2$. Let $v \in H$ be a vertex included in H. It holds that $|N_G^2[v]| > 10$. But since $\Delta(G) \leq 3$, it also holds that

$$|N_G^2[v]| \leq 1 + \sum_{u \in N_G(v)} |N_G[u] \setminus \{v\}| \leq 1 + \sum_{u \in N_G(v)} 3 \leq 10.$$

This is a contradiction. □

Clearly, Reduction Rule 3.14 can be applied in linear time. Let $q := \max_{H \in \mathcal{H}} |H|$ be the size of the largest habitat. After the application of Reduction Rule 3.14, it holds that $q \leq 10$. We recall two reduction rules.

Reduction Rule 3.3 (Restated)**.** *If for an edge $e \in E(G)$ there is no habitat $H \in \mathcal{H}$ with $e \subseteq H$, then delete e.*

Reduction Rule 3.9 (Restated)**.** *If for an edge $e \in E(G)$ there is a habitat $H \in \mathcal{H}$ with $e \subseteq H$ such that e is not contained in a triangle in G, then fix $e \in F$.*

Since $q \leq 10$, both Reduction Rules 3.3 and 3.9 can be exhaustively applied in linear time by Observations 3.4 and 3.10. The following holds after exhaustive application of Reduction Rules 3.3 and 3.9.

Observation 3.16. *Every edge of G is contained in a triangle in G or fixed to be included in the solution F.*

Therefore, if we choose for one triangle $Q \subseteq G$ which edges of Q to add to F, then this choice may only influence which edges of any other triangle contained in G need to be added to F. Due to $\Delta(G) \leq 3$, the triangles in G are not well connected. Ideally, there is a set $\mathcal{S}$ of small subgraphs of G with the properties that (1) each triangle of G is included in one of these subgraphs and (2) for each subgraph $S \in \mathcal{S}$ the choice of which edges of S to add to F can be made independently. Based on this, we will first introduce the concept of *super-triangles* and then the concept of *areas*.

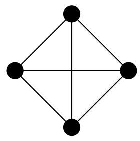
As a preparation, we remove some subgraphs of G that would hinder our approach. If a graph G' is isomorphic to the graph depicted in Figure 3.1a$_1$, then we refer to G' as a $G_4^{\times}$. If a graph G' is isomorphic to the graph depicted in Figure 3.1a$_2$, then we refer to G' as a $G_6^{\times}$. We exhaustively apply Reduction Rule 3.11, which has the effect of deleting all components of G with a vertex set of size at most six. This can be done in linear time due to $q \leq 10$ and Observation 3.12. After the exhaustive application of Observation 3.12, the following holds.

Observation 3.17. *The graph G contains no $G_4^{\times}$ and no $G_6^{\times}$.*

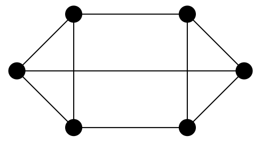
If a graph G' is isomorphic to the graph depicted in Figure 3.1b$_1$, then we refer to G' as a $G_4^{\checkmark}$. If a graph G' is isomorphic to the graph depicted in Figure 3.1b$_2$, then we refer to G' as a $G_6^{\checkmark}$.

Definition 3.18. We call a subgraph $S \subseteq G$ of G a *super-triangle* if S is a $G_4^{\checkmark}$ or a $G_6^{\checkmark}$.

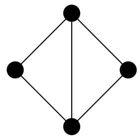
For our idea to work, it is important that the edge sets of super-triangles do not intersect. This is what we show next.



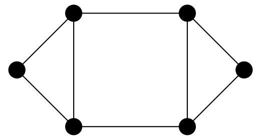
(a$_1$) Depiction of a $G_4^{\times}$.



(a$_2$) Depiction of a $G_6^{\times}$.

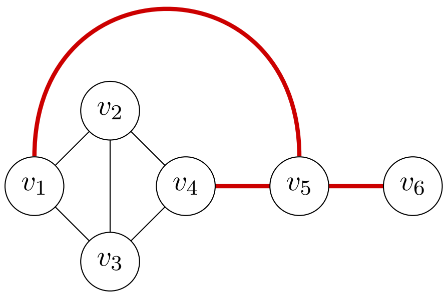


(b$_1$) Depiction of a $G_4^{\checkmark}$.

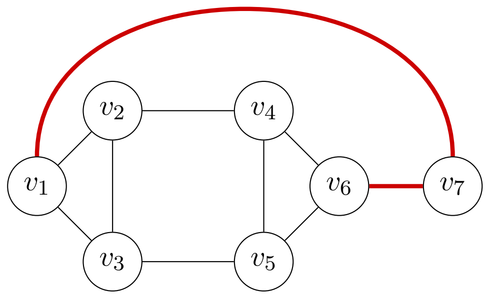


(b$_2$) Depiction of a $G_6^{\checkmark}$.

Figure 3.1: (a$_1$)&(a$_2$) None of these two graphs is included as a subgraph in G (Observation 3.17). (b$_1$)&(b$_2$) A subgraph $S \subseteq G$ of G is a super-triangle if and only if S is a $G_4^{\checkmark}$ or a $G_6^{\checkmark}$ (Definition 3.18).



(a)



(b)

Figure 3.2: Illustration to the proof of Observation 3.19. The edges of S_1 are black and thin. The edge set of S_2 contains some of the black thin edges and all red thick edges. (a) First case, i.e., S_1 is a $G_4^{\checkmark}$. (b) Second case, i.e., S_1 is a $G_6^{\checkmark}$.

Observation 3.19. *Let $S_1, S_2 \subseteq G$ be two super-triangles. Then, it holds that $E(S_1) \cap E(S_2) = \emptyset$.*

Proof. Assume towards a contradiction that there are two super-triangles $S_1, S_2 \subseteq G$ with $E(S_1) \cap E(S_2) \neq \emptyset$. We distinguish two cases.

First case: Super-triangle S_1 is a $G_4^{\checkmark}$ (see Figure 3.2a). Let $V(S_1) = \{v_1, \ldots, v_4\}$ and $E(S_1) = [V(S_1)]^2 \setminus \{\{v_1, v_4\}\}$. We try to derive the structure of S_2. Since G contains no $G_4^{\times}$, it holds that $\{v_1, v_4\} \notin E(G)$. It follows that $V(S_2) \nsubseteq V(S_1)$. Hence, w.l.o.g. (by symmetry) there is a vertex $v_5 \in V(G)$ such that $\{v_4, v_5\} \in E(S_2)$. As every super-triangle is 2-connected, there is a (v_4, v_5)-path $P \subseteq S_2$ such that $\{v_4, v_5\} \notin E(P)$. From $\Delta(G) \leq 3$ it follows that there is a (v_1, v_5)-path $P \subseteq S_2$ with $\{v_4, v_5\} \notin E(P)$. Since every super-triangle's longest chordless cycle has length at most four, it follows that $\{v_1, v_5\} \in E(S_2)$. Observe that $V(S_2) \nsubseteq V(S_1) \cup \{v_5\}$ must hold. Hence, there is a vertex $v_6 \in V(G)$ with $\{v_5, v_6\} \in E(S_2)$. Because of $\Delta(G) \leq 3$, the vertex set $\{v_5\}$ separates S_2. Thus, S_2 is not 2-connected, a contradiction to S_2 being a super-triangle.

Second case: Super-triangle S_1 is a $G_6^{\checkmark}$ (see Figure 3.2b). Let $V(S_1) = \{v_1, \ldots, v_6\}$ and $E(S_1) = [\{v_1, v_2, v_3\}]^2 \cup [\{v_4, v_5, v_6\}]^2 \cup \{\{v_2, v_4\}, \{v_3, v_5\}\}$. We try to derive the structure of S_2. Since G contains no $G_6^{\times}$, it holds that $\{v_1, v_6\} \notin E(G)$. It follows that $V(S_2) \nsubseteq V(S_1)$. Hence, w.l.o.g. (by symmetry) there is a vertex $v_7 \in V(G)$ such that $\{v_6, v_7\} \in E(S_2)$. As every super-triangle is 2-connected, there is a (v_6, v_7)-path $P \subseteq S_2$ such that $\{v_6, v_7\} \notin E(P)$. From $\Delta(G) \leq 3$ it follows that there is a (v_1, v_7)-path $P \subseteq S_2$ with $\{v_6, v_7\} \notin E(P)$. Therefore, S_2 contains a chordless cycle of length at least five. Because every super-triangle's longest chordless cycle has length at most four, this is a contradiction to S_2 being a super-triangle. $\square$

The following definition is useful because every triangle in G is included in an area.

Definition 3.20. We call a subgraph $A \subseteq G$ an *area* if

- A is a super-triangle or
- A is a triangle and there is no super-triangle $S \subseteq G$ with $A \subseteq S$.

We show that the edge sets of all areas in G are pairwise disjoint.

Observation 3.21. *Let $A_1, A_2 \subseteq G$ be two distinct areas. Then, it holds that $E(A_1) \cap E(A_2) = \emptyset$.*

Proof. We distinguish three cases.

First case: The areas A_1 and A_2 are both super-triangles. Then, the statement follows from Observation 3.19.

Second case: The areas A_1 and A_2 are both triangles. Assume towards a contradiction that $E(A_1) \cap E(A_2) \neq \emptyset$. Then, $A_1 \cup A_2$ is a $G_4^{\checkmark}$, a contradiction to A_1 and A_2 being distinct areas.

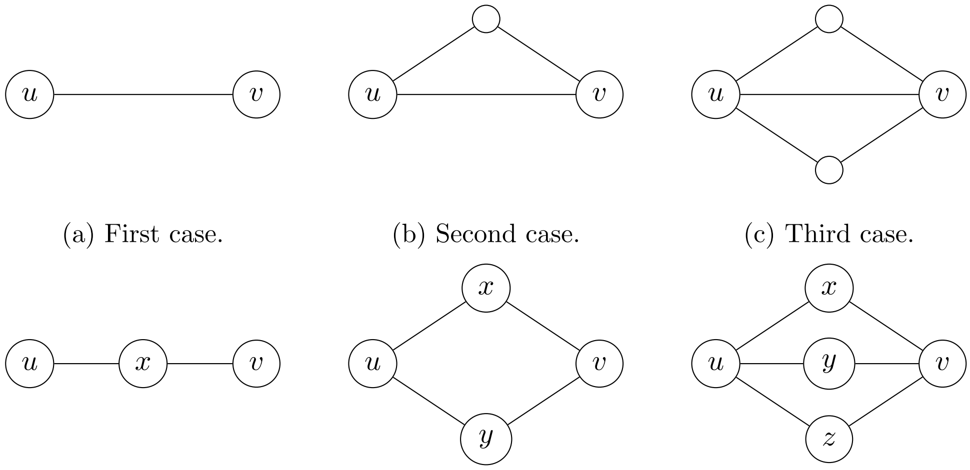
Third case: The area A_1 is a triangle and the area A_2 is a super-triangle. Since A_2 contains no edge $\{u, v\} \in E(A_2)$ such that $\deg_{A_2}(u) \leq 2$ and $\deg_{A_2}(v) \leq 2$, it follows that $|E(A_1) \cap E(A_2)| \neq 1$. As G contains no $G_4^{\times}$ and no $G_6^{\times}$, it holds that $|E(A_1) \cap E(A_2)| \neq 2$. From the definition of areas it follows that $|E(A_1) \cap E(A_2)| \neq 3$. Thus, $E(A_1) \cap E(A_2) = \emptyset$. $\square$

Before moving on to the main algorithm, we apply one last reduction rule. It ensures that a solution F always exists. (After the application of the reduction rule, the set $E(G)$ is a solution.)

Reduction Rule 3.22. *If there is a habitat $H \in \mathcal{H}$ with $\operatorname{diam}(G[H]) > 2$, then return a trivial* `no`*-instance.*

The correctness directly follows from the definition of 2-Diam GBP. Since for every habitat $H \in \mathcal{H}$ it holds that $|H| \leq 10$, the graph $G[H]$ can be constructed in constant time and the diameter of $G[H]$ can also be computed in constant time. This yields an overall linear running time for Reduction Rule 3.22.

Next, we describe the main algorithm. Let F_{fixed} be the set of edges fixed to be included in the solution F during the application of Reduction Rule 3.9. For every area $A \subseteq G$ and every habitat $H \in \mathcal{H}$, let $\bar{E}_{H,A} := E(G[H]) \setminus E(A)$ be the set of *outside edges* of A with respect to H.



(a) First case. (b) Second case. (c) Third case.

(d) Fourth case. (e) Fifth case. (f) Sixth case.

Figure 3.3: Illustration of the cases occurring in the case distinction in Lemma 3.24.

Algorithm 1 Algorithm for deciding pre-processed instances of 2-Diam GBP-C on graphs of maximum degree at most three.

Let F be initially equal to F_{fixed}. For each area $A \subseteq G$ find a minimum cost edge set $F_A \subseteq E(A)$ with $F_{\text{fixed}} \cap E(A) \subseteq F_A$ such that for all habitats $H \in \mathcal{H}$ with $H \cap V(A) \neq \emptyset$ it holds that $\operatorname{diam}(G[\bar{E}_{H,A} \cup F_A][H]) \leq 2$ and add F_A to F. If $c(F) \leq k$, then return `yes`. Otherwise, return `no`.

We remark that during the application of the algorithm a set F_A can always be found because we have previously applied Reduction Rule 3.22. In the following let F be the solution constructed by Algorithm 1. For each area $A \subseteq G$, let F_A be the edge set that Algorithm 1 has found for A. Moreover, let $F_{H,A} := \bar{E}_{H,A} \cup F_A$.

Observation 3.23. *Let $H \in \mathcal{H}$ be a habitat and let $u, v \in H$ be two distinct vertices. Let $\mathcal{P}^{u,v}_{\leq 2}$ be the set of (u,v)-paths of length at most two in $G[H]$. If there is an area $A \subseteq G$ with $P \subseteq A$ for all $P \in \mathcal{P}^{u,v}_{\leq 2}$, then it holds that $\operatorname{dist}_{G[F][H]}(u,v) \leq 2$.*

Proof. By definition of Algorithm 1, it holds that $\operatorname{diam}(G[F_{H,A}][H]) \leq 2$. Since $F_A \subseteq F$, this implies that $\operatorname{diam}(G[\bar{E}_{H,A} \cup F][H]) \leq 2$. It follows that $\operatorname{dist}_{G[\bar{E}_{H,A} \cup F][H]}(u,v) \leq 2$. As $E(P) \cap \bar{E}_{H,A} = \emptyset$ for every $P \in \mathcal{P}^{u,v}_{\leq 2}$, it follows that $\operatorname{dist}_{G[F][H]}(u,v) \leq 2$. □

Lemma 3.24. *The set F constructed by Algorithm 1 is a solution.*

Proof. Let $H \in \mathcal{H}$ be a habitat and let $u, v \in H$ be vertices included in H. It suffices to show that $\operatorname{dist}_{G[F][H]}(u,v) \leq 2$. Because of Reduction Rule 3.22, it holds that $\operatorname{dist}_{G[H]}(u,v) \leq 2$. Hence, there is a at least one (u,v)-path of length at most two in $G[H]$. Let $\mathcal{P}^{u,v}_{\leq 2}$ be the set of (u,v)-paths of length at most two in $G[H]$. We do a case distinction on $\mathcal{P}^{u,v}_{\leq 2}$. For a depiction of all six cases, see Figure 3.3.

First case: The set $\mathcal{P}^{u,v}_{\leq 2}$ contains one path of length 1 and no path of length 2 (see Figure 3.3a). Then, the edge $\{u, v\}$ is not part of a triangle in $G[H]$. Because of Observation 3.16, it follows that $\{u, v\} \in F$. Hence, $\operatorname{dist}_{G[F][H]}(u,v) \leq 2$.

Second case: The set $\mathcal{P}^{u,v}_{\leq 2}$ contains one path of length 1 and one path of length 2 (see Figure 3.3b). Then, the union of the two paths in $\mathcal{P}^{u,v}_{\leq 2}$ is a triangle. Since each triangle is contained in an area, it follows that there is an area $A \subseteq G$ with $P \subseteq A$ for every path $P \in \mathcal{P}^{u,v}_{\leq 2}$. Using Observation 3.23 we get $\mathrm{dist}_{G[F][H]}(u, v) \leq 2$.

Third case: The set $\mathcal{P}^{u,v}_{\leq 2}$ contains one path of length 1 and two paths of length 2 (see Figure 3.3c). Then, the union of the three paths in $\mathcal{P}^{u,v}_{\leq 2}$ is a $G^{\checkmark}_4$. Hence, there again is an area $A \subseteq G$ with $P \subseteq A$ for every path $P \in \mathcal{P}^{u,v}_{\leq 2}$. Using Observation 3.23 we get $\mathrm{dist}_{G[F][H]}(u, v) \leq 2$.

Fourth case: The set $\mathcal{P}^{u,v}_{\leq 2}$ contains no path of length 1 and one path of length 2. Let $x \in H$ be the unique vertex with $\{u, x\}, \{x, v\} \in E(G)$ (see Figure 3.3d). Assume towards a contradiction that $\mathrm{dist}_{G[F][H]}(u, v) > 2$. W.l.o.g. (by symmetry) it holds that $\{u, x\} \notin F$. Because of Observation 3.16, there is a triangle $Q \subseteq G[H]$ such that $\{u, x\} \in E(Q)$. As each triangle is contained in an area, it follows that there is an area $A \subseteq G$ with $Q \subseteq A$. By definition of Algorithm 1, it holds that $\mathrm{diam}(G[F_{H,A}][H]) \leq 2$. Hence, $\mathrm{dist}_{G[F_{H,A}][H]}(u, v) \leq 2$. Since the path $P \in \mathcal{P}^{u,v}_{\leq 2}$ is the only (u, v)-path of length at most two in $G[H]$, it follows that $\{u, x\} \in F$, a contradiction.

Fifth case: The set $\mathcal{P}^{u,v}_{\leq 2}$ contains no path of length 1 and two paths of length 2. Let $x, y \in H$ be the two distinct vertices with $\{u, x\}, \{x, v\} \in E(G)$ and $\{u, y\}, \{y, v\} \in E(G)$ (see Figure 3.3e). Since $\mathcal{P}^{u,v}_{\leq 2}$ contains no path of length one, it holds that $\{u, v\} \notin E(G)$. We distinguish between the cases that $\{x, y\} \in E(G)$ and $\{x, y\} \notin E(G)$. First, we consider the case that $\{x, y\} \in E(G)$. Then, the graph $A := G[\{u, v, x, y\}]$ is a $G^{\checkmark}_4$. Therefore, A is an area. As for every $P \in \mathcal{P}^{u,v}_{\leq 2}$ it holds that $P \subseteq A$, it follows from Observation 3.23 that $\mathrm{dist}_{G[F][H]}(u, v) \leq 2$. This concludes the proof for the case that $\{x, y\} \in E(G)$.

Now let $\{x, y\} \notin E(G)$. Assume towards a contradiction that $\mathrm{dist}_{G[F][H]}(u, v) > 2$. Then, it holds w.l.o.g. (by symmetry) that $\{u, x\}, \{u, y\} \notin F$ (see Figure 3.4a) or $\{u, x\}, \{y, v\} \notin F$ (see Figure 3.4c).

Let us consider the case that $\{u, x\}, \{u, y\} \notin F$. Because of Observation 3.16, each of the edges $\{u, x\}$ and $\{u, y\}$ is contained in a triangle in $G[H]$. Due to $\Delta(G) \leq 3$ and $\{u, v\}, \{x, y\} \notin E(G)$, it follows that there exists a vertex $z \in H$ such that $\{u, z\}, \{x, z\}, \{y, z\} \in E(G)$ (see Figure 3.4b). The graph $A := G[\{u, x, y, z\}]$ is a $G^{\checkmark}_4$. Therefore, A is an area. By definition of Algorithm 1, it holds that $\mathrm{diam}(G[F_{H,A}][H]) \leq 2$. Hence, $\mathrm{dist}_{G[F_{H,A}][H]}(u, v) \leq 2$. But since each (u, v)-path of length at most two in $G[H]$ contains the edge $\{u, x\}$ or $\{u, y\}$, it holds that $\mathrm{dist}_{G[F_{H,A}][H]}(u, v) > 2$, a contradiction.

Finally, we look at the case that $\{u, x\}, \{y, v\} \notin F$. Because of Observation 3.16, each of the edges $\{u, x\}$ and $\{y, v\}$ is contained in a triangle in $G[H]$. Due to $\Delta(G) \leq 3$ and $\{u, v\}, \{x, y\} \notin E(G)$, it follows that there are distinct vertices $z_1, z_2 \in H$ such that $\{u, z_1\}, \{x, z_1\} \in E(G)$ and $\{y, z_2\}, \{v, z_2\} \in E(G)$ (see Figure 3.4d). The graph $A := G[\{u, v, x, y, z_1, z_2\}]$ is a $G^{\checkmark}_6$. Therefore, A is an area. By definition of Algorithm 1, it holds that $\mathrm{diam}(G[F_{H,A}][H]) \leq 2$. Hence, $\mathrm{dist}_{G[F_{H,A}][H]}(u, v) \leq 2$. But since each (u, v)-path of length at most two in $G[H]$ contains the edge $\{u, x\}$ or $\{y, v\}$, it holds that $\mathrm{dist}_{G[F_{H,A}][H]}(u, v) > 2$, a contradiction.

Sixth case: The set $\mathcal{P}^{u,v}_{\leq 2}$ contains no path of length 1 and three paths of length 2. Let $x, y, z \in H$ be the three mutually distinct vertices with $\{u, w\}, \{w, v\} \in E(G)$

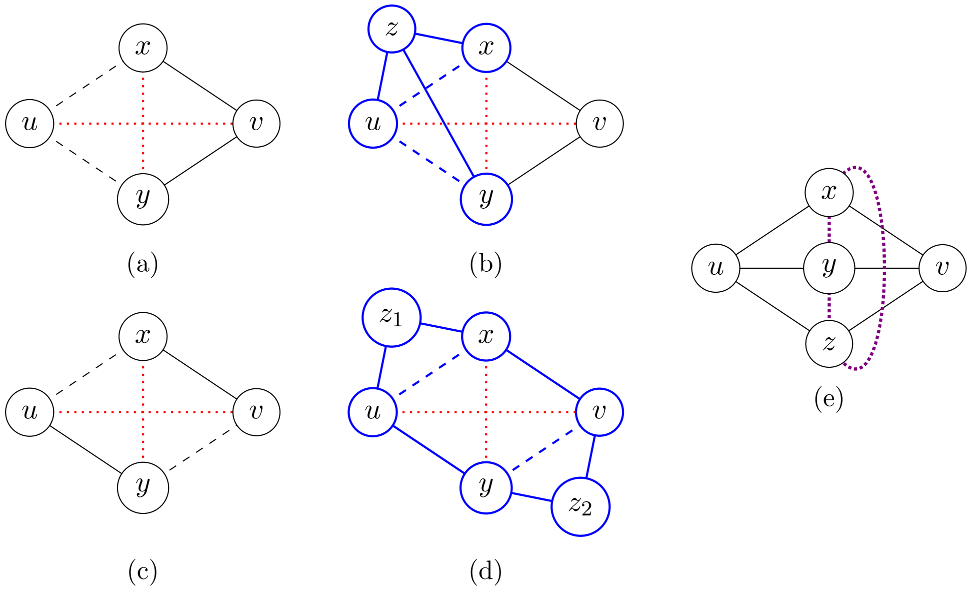


Figure 3.4: Illustration of the fifth and sixth case in the proof of Lemma 3.24. Solid edges are included in $E(G)$ and may be included in F. Dashed edges are included in $E(G)$ but not included in F. Red dotted edges are not included in $E(G)$. (a)&(b) Fifth case, subcase where $\{u, x\}, \{u, y\} \notin F$. (b) The blue thick subgraph is a $G_4^{\checkmark}$. (c)&(d) Fifth case, subcase where $\{u, x\}, \{y, v\} \notin F$. (d) The blue thick subgraph is a $G_6^{\checkmark}$. (e) Sixth case, only one of the three violet densely dotted edges can be included in $E(G)$.

for each $w \in \{x, y, z\}$ (see Figure 3.3f). Because of $\Delta(G) \leq 3$, at most one of the edges $\{x, y\}, \{y, z\}, \{x, z\}$ is contained in $E(G)$ (see Figure 3.4e). Thus, there is a path $P \in \mathcal{P}_{\leq 2}^{u,v}$ such that for both edges $e \in E(P)$ it holds that e is not contained in a triangle in G. Due to Observation 3.16, it follows that $E(P) \subseteq F$. Hence, $\text{dist}_{G[F][H]}(u, v) \leq 2$. □

Lemma 3.25. *No solution of lower cost than $c(F)$ exists.*

Proof. Assume towards a contradiction that there is a solution F' with $c(F') < c(F)$. Since Observation 3.16 only fixes edges that are included in every solution, it holds that $F_{\text{fixed}} \subseteq F'$. Thus, F and F' only differ by edges that are contained in triangles in G. As each triangle is contained in an area, there is an area $A \subseteq G$ with $c(E(A) \cap F') < c(E(A) \cap F)$. As the edge sets of distinct areas do not intersect (Observation 3.21), it holds that $F_A = E(A) \cap F$. Let $F'_A := E(A) \cap F'$. Then, it holds that $c(F'_A) < c(F_A)$, a contradiction to Algorithm 1 selecting F_A such that the cost is minimum. □

The correctness of Algorithm 1 follows from Lemmas 3.24 and 3.25. It remains to show that Algorithm 1 runs in linear time. Since $\Delta(G) \leq 3$, there is a constant that bounds the size of the 3-neighborhood of every vertex $v \in V(G)$ in G. Hence, for every vertex $v \in V(G)$ all areas $A \subseteq G$ with $v \in V(A)$ can be found in constant time. Therefore, it takes $\mathcal{O}(|V(G)|)$ time to find all areas $A \subseteq G$. Because of $\Delta(G) \leq 3$, $q \leq 10$ and because the edge sets of distinct areas do not intersect (Observation 3.21), there are

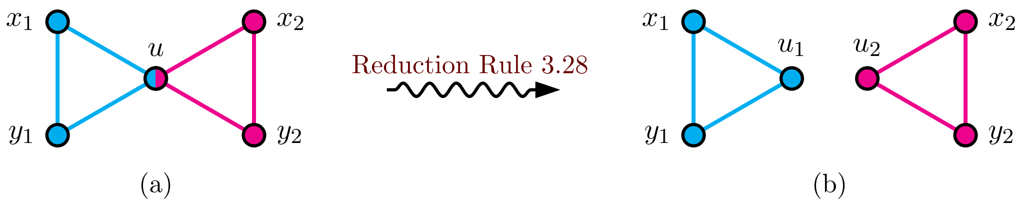


Figure 3.5: Illustration to Reduction Rule 3.28. Subgraphs induced by habitats containing the vertex u or one of the vertices u_1, u_2 are marked by differing colors. (a) The butterfly graph $S \subseteq G$ selected in the application of the reduction rule. (b) The situation created by the reduction rule.

no more than constantly many areas $A \subseteq G$ with $H \cap V(A)$ for each habitat $H \in \mathcal{H}$. It follows that the number of area-habitat-pairs $A \subseteq G$, $H \in \mathcal{H}$ with $H \cap V(A) \neq \emptyset$ is in $\mathcal{O}(|V(G)|)$. This allows for an overall linear running time. Thus, the proof of Proposition 3.13 is finished.

3.2.2 A Polynomial-Time Algorithm for $\Delta \leq 4$

In this section, we show the following.

Proposition 3.26. 2-Diam GBP-C *can be solved in $\mathcal{O}(|V(G)|^2 \cdot \log |V(G)|)$ time on graphs of maximum degree at most four if each habitat induces a triangle.*

The following problem can be solved in $\mathcal{O}(|V(G)|^2 \cdot \log |V(G)|)$ time on claw-free graphs [NS21].

Problem: Weighted Vertex Cover
Input: A vertex-weighted graph (G, w) and an integer $k \in \mathbb{N}$.
Question: Is there a vertex cover $S \subseteq V(G)$ with $w(S) \leq k$?

We give a reduction to Weighted Vertex Cover on claw-free graphs. Let $\mathcal{I} = (G, c, \mathcal{H}, k)$ be an instance of 2-Diam GBP-C with maximum degree four and each habitat inducing a triangle. Because of $\Delta(G) \leq 4$, it holds that each edge is contained in at most three triangles in G. Moreover, the number of edges in G is in $\mathcal{O}(|V(G)|)$. Since every habitat induces a triangle, the number of habitats is also in $\mathcal{O}(|V(G)|)$. Therefore, a running time linear in the input size $|\mathcal{I}|$ is also linear in $|V(G)|$. As we will see, the reduction to Weighted Vertex Cover only takes $\mathcal{O}(|V(G)|)$ time. Thus, the asymptotic running time bound of the overall algorithm only depends on the time needed for solving the produced instance of Weighted Vertex Cover.

Before preceding with the actual reduction to Weighted Vertex Cover, we do some preprocessing on $\mathcal{I}$. First, we exhaustively apply Reduction Rule 3.3 to delete all non-habitat edges. By Observation 3.10, this takes $|V(G)|$ time. As each habitat induces a triangle, each remaining edge of G is included in a triangle in G.

Definition 3.27. A graph is a *butterfly graph* if it is isomorphic to the graph depicted in Figure 3.5a. The vertex of degree four is called the *center* of the butterfly graph.

Next, we exhaustively apply the following reduction rule. Doing this removes all induced butterfly graphs.

Reduction Rule 3.28. *If G contains a butterfly graph $S \subseteq G$ as an induced subgraph, then do the following (see Figure 3.5 for an illustration). Let $V(S) = \{u, x_1, y_1, x_2, y_2\}$ and $E(S) = [\{u, x_1, y_1\}]^2 \cup [\{u, x_2, y_2\}]^2$. Delete u and all edges incident to u. Add two vertices called u_1 and u_2. Moreover, add the edges $\{u_1, x_1\}$, $\{u_1, y_1\}$, $\{u_2, x_2\}$, and $\{u_2, y_2\}$ with the edge costs of the new edges mirroring the edge costs of the deleted edges, i.e., $c(\{u_1, x_1\}) := c(\{u, x_1\})$, etc. If $\{u, x_1, y_1\} \in \mathcal{H}$, then delete $\{u, x_1, y_1\}$ from $\mathcal{H}$ and add $\{u_1, x_1, y_1\}$ to $\mathcal{H}$ instead. Likewise, if $\{u, x_2, y_2\} \in \mathcal{H}$, then delete $\{u, x_2, y_2\}$ from $\mathcal{H}$ and add $\{u_2, x_2, y_2\}$ to $\mathcal{H}$.*

Observation 3.29. *Reduction Rule 3.28 is correct.*

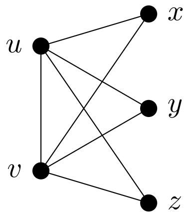
Proof. Let $\mathcal{I}' = (G', c', \mathcal{H}', k')$ be the instance of 2-Diam GBP-C obtained from $\mathcal{I}$ by application of Reduction Rule 3.28. We show that $\mathcal{I}$ is a `yes`-instance if and only if $\mathcal{I}'$ is a `yes`-instance. Let $f : E(G) \to E(G')$ be a function such that for every $e \in E(G)$ it holds that

$$f(e) := \begin{cases} \{u_1, x_1\}, & \text{if } e = \{u, x_1\}, \\ \{u_1, y_1\}, & \text{if } e = \{u, y_1\}, \\ \{u_2, x_2\}, & \text{if } e = \{u, x_2\}, \\ \{u_2, y_2\}, & \text{if } e = \{u, y_2\}, \\ e, & \text{otherwise.} \end{cases}$$

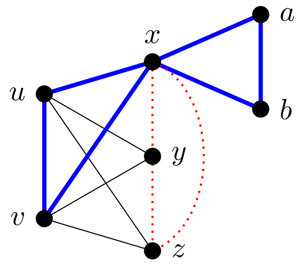
($\Rightarrow$) Let F be a solution to $\mathcal{I}$. We claim that $F' := \{f(e) \mid e \in F\}$ is a solution to $\mathcal{I}'$. Note that $|F'| \le k'$. For every habitat $H \in \mathcal{H}'$ with $H \cap \{u_1, u_2\} = \emptyset$, it holds that $\operatorname{diam}(G'[F'][H]) \le 2$. Since each habitat induces a triangle, there are at most two habitats in $\mathcal{H}'$ that intersect with $\{u_1, u_2\}$. These intersecting habitats are $\{u_1, x_1, y_1\}$ and $\{u_2, x_2, y_2\}$. Assume that $\{u_1, x_1, y_1\} \in \mathcal{H}'$. Then, it holds that $\{u, x_1, y_1\} \in \mathcal{H}$. Therefore, $\operatorname{diam}(G[F][\{u, x_1, y_1\}]) \le 2$. By construction of $\mathcal{I}'$, this implies that $\operatorname{diam}(G'[F'][\{u_1, x_1, y_1\}]) \le 2$.

($\Leftarrow$) Let F' be a solution to $\mathcal{I}'$ and let $f^{-1} : E(G') \to E(G)$ be the inverse function of f. We claim that $F := \{f^{-1}(e) \mid e \in F'\}$ is a solution to $\mathcal{I}$. Note that $|F| \le k$. For every habitat $H \in \mathcal{H}$ with $u \notin H$, it holds that $\operatorname{diam}(G[F][H]) \le 2$. Since $\Delta(G) \le 4$ and no edge from $\{\{z_1, z_2\} \mid z_1 \in \{x_1, y_1\} \land z_2 \in \{x_2, y_2\}\}$ is included in $E(G)$, there are at most two habitats in $\mathcal{H}$ that contain u. These habitats are $\{u, x_1, y_1\}$ and $\{u, x_2, y_2\}$. Assume that $\{u, x_1, y_1\} \in \mathcal{H}$. Then, it holds that $\{u_1, x_1, y_1\} \in \mathcal{H}'$. Therefore, $\operatorname{diam}(G'[F'][\{u_1, x_1, y_1\}]) \le 2$. By construction of $\mathcal{I}'$, this implies that $\operatorname{diam}(G[F][\{u, x_1, y_1\}]) \le 2$. $\square$

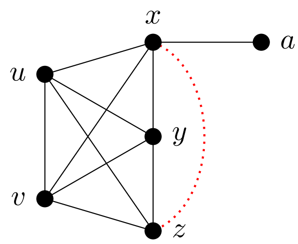
Reduction Rule 3.28 can be applied exhaustively in linear time the following way. For each vertex $u \in V(G)$ we check whether it is the center of an induced butterfly graph. This only requires examining the neighbors of u and can be done in constant time for each single $u \in V(G)$ because of $\Delta(G) \le 4$. Hence, finding all center vertices of induced butterfly graphs takes $\mathcal{O}(|V(G)|)$ time. Since each application of Reduction Rule 3.28 removes one center vertex of an induced butterfly graph and does not create new center vertices of induced butterfly graphs, we only need to run Reduction Rule 3.28



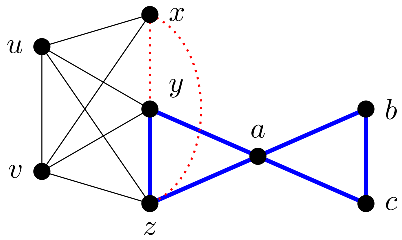
(a) "Base Situation".



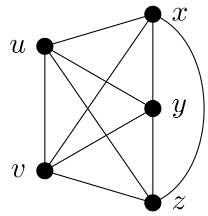
(b) First case.



(d) Third case.



(c) Second case.



(e) Fourth case.

Figure 3.6: Illustration to the proof of Observation 3.30. Red dotted edges are not included in $E(G)$. The subgraphs marked in blue are induced butterfly graphs. (a) An edge $\{u, v\} \in E(G)$ that is contained in three triangles in G. (b)–(e) Illustration of the four cases distinguished in the proof of Observation 3.30.

$\mathcal{O}(|V(G)|)$ times. As we have already identified all center vertices, a single application of Reduction Rule 3.28 requires only constant time. Thus, all together can be done in $\mathcal{O}(|V(G)|)$ time.

Moreover, we exhaustively apply Reduction Rule 3.11, which has the effect of deleting all components of G with a vertex set of size at most six. By Observation 3.12, the time needed for this also is in $\mathcal{O}(|V(G)|)$. Having carried out the reductions, we can make the following observation.

Observation 3.30. *Each edge of G is contained in at most two triangles in G.*

Proof. Assume towards a contradiction that there is an edge $\{u, v\} \in E(G)$ such that $\{u, v\}$ is contained in three triangles in G. Hence, there are three vertices $x, y, z \in V(G)$ such that each of the sets $\{u, v, x\}$, $\{u, v, y\}$, $\{u, v, z\}$ induces a triangle in G (see Figure 3.6a). Let $\bar{V} := V(G) \setminus \{u, v, x, y, z\}$. Since every component in G has more than six vertices and $\Delta(G) \leq 4$, there is a vertex $a \in \bar{V}$ such that a is adjacent to a vertex in $\{x, y, z\}$. We do a case distinction on the size of the edge set $E_{xyz} := [\{x, y, z\}]^2 \cap E(G)$.

First case: It holds that $|E_{xyz}| = 0$ (see Figure 3.6b). Assume w.l.o.g. (by symmetry) that a is adjacent to x. As every edge is contained in a triangle in G, there is a vertex $b \in \bar{V}$ such that $G[\{x, a, b\}]$ is a triangle. But then $G[\{u, v, x, a, b\}]$ is an induced butterfly graph in G, a contradiction.

Second case: It holds that $|E_{xyz}| = 1$ (see Figure 3.6c). Assume w.l.o.g. (by symmetry) that $\{y, z\} \in E_{xyz}$. By argumentation analogous to the first case, the vertex a is not adjacent to x. (And the only vertices adjacent to x are u and v.) Thus, assume

Figure 3.7: Illustration to Construction 3.31. (a) Example graph G of the input instance $\mathcal{I} = (G, c, \mathcal{H}, k)$. The subgraphs induced by the habitats in $\mathcal{H}$ are marked by differing colors. (b) Depicted in black, the graph G' obtained from G using Construction 3.31. For illustration, the graph G of the input instance is faintly visible in the background.

w.l.o.g. (by symmetry) that a is adjacent to y. As every edge is contained in a triangle in G and because G contains no butterfly graph as an induced subgraph, it holds that a is also adjacent to z. Since every component in G has more than six vertices, there is a vertex $b \in \bar{V} \setminus \{a\}$ such that b is adjacent to a. As, again, every edge is contained in a triangle in G, this implies that is a further vertex $c \in \bar{V}$ such that $G[\{a, b, c\}]$ is a triangle. But then $G[\{y, z, a, b, c\}]$ is an induced butterfly graph in G, a contradiction.

Third case: It holds that $|E_{xyz}| = 2$ (see Figure 3.6d). Assume w.l.o.g. (by symmetry) that $\{x, y\}, \{y, z\} \in E_{xyz}$. Moreover, assume w.l.o.g. (by symmetry) that a is adjacent to x. Then, it holds that $\{x, a\}$ is not contained in a triangle in G, a contradiction.

Fourth case: It holds that $|E_{xyz}| = 3$ (see Figure 3.6e). Then, $G[\{u, v, x, y, z\}]$ is a component of G containing less than six vertices, a contradiction. □

Finally, we reduce the preprocessed instance to WEIGHTED VERTEX COVER. Note that executing the following construction takes $|\mathcal{O}(|V(G)|)|$ time.

Construction 3.31. Let $\mathcal{I} = (G, c, \mathcal{H}, k)$ be an instance of 2-DIAM GBP-C preprocessed as described above. Construct an instance $\mathcal{I}' = (G', w, k')$ of WEIGHTED VERTEX COVER with $k' := k$ as follows (see Figure 3.7 for an illustration).

Let G' be initially empty. For each edge $e \in E(G)$, add a vertex v_e to $V(G')$ and set $w(v_e) := c(e)$. For each habitat $\{x, y, z\} \in \mathcal{H}$, add the edges $\{v_{\{x,y\}}, v_{\{x,z\}}\}$, $\{v_{\{x,y\}}, v_{\{y,z\}}\}$, and $\{v_{\{x,z\}}, v_{\{y,z\}}\}$ to $E(G')$.

Observation 3.32. *The graph G' constructed in Construction 3.31 is claw-free.*

Proof. Assume towards a contradiction that G' contains an induced claw $C \subseteq G'$. Let $v_e \in V(C)$ be the center vertex of C. Since v_e is the center of an induced claw, there is a three-element set $N_{\text{ind}} \subseteq N_{G'}(v_e)$ of neighbors of v_e in G' such that $[N_{\text{ind}}]^2 \cap E(G') = \emptyset$. Let $v_f, v_g, v_h \in N_{\text{ind}}$ be the three elements of N_{ind}. By construction of $\mathcal{I}'$, it follows that for each habitat $H \in \mathcal{H}$ at most one of the three edges f, g, and h is contained in the edge set of $G[H]$. By Observation 3.30, it holds that e is contained in at most two triangles in G. Let $\mathcal{H}_e \subseteq \mathcal{H}$ be the set containing every habitat $H \in \mathcal{H}$ with the property that e is contained in the edge set of $G[H]$. It holds that $|\mathcal{H}_e| \leq 2$. As v_f, v_g, and v_h are neighbors of v_e in G', it follows that for each edge $e_N \in \{f, g, h\}$ there is a habitat $H \in \mathcal{H}_e$ such that e_N is contained in the edge set of $G[H]$. By pigeonhole principle using

$|\mathcal{H}_e| \leq 2$, it follows that there is a habitat $H \in \mathcal{H}_e$ such that at least two of the three edges f, g, and h are contained in the edge set of $G[H]$, a contradiction. □

Lemma 3.33. *Let $\mathcal{I}'$ be the instance of* Weighted Vertex Cover *obtained from a preprocessed instance $\mathcal{I}$ of* 2-Diam GBP-C *using Construction 3.31. Then, $\mathcal{I}$ is a* `yes`*-instance if and only if $\mathcal{I}'$ is a* `yes`*-instance.*

Proof. ($\Rightarrow$) Let F be a solution to $\mathcal{I}$. We claim that $S := \{v_e \mid e \in F\}$ is a vertex cover of G'. Note that $\sum_{v \in S} w(v) \leq k'$. Assume towards a contradiction that there is an edge $\{v_e, v_f\} \in E(G')$ with $v_e, v_f \notin S$. It follows that $e, f \notin F$. By construction of $\mathcal{I}'$, there is a habitat $H \in \mathcal{H}$ with $e, f \in E(G[H])$. From $e, f \notin F$ it follows that $\operatorname{diam}(G[F][H]) = \infty$, a contradiction to F being a solution to $\mathcal{I}$.

($\Leftarrow$) Let $S \subseteq V(G')$ be a vertex cover of G' with $\sum_{v \in S} w(v) \leq k'$. We claim that $F := \{e \mid v_e \in S\}$ is a solution to $\mathcal{I}$. Note that $c(F) \leq k$. Assume towards a contradiction that there is a habitat $H \in \mathcal{H}$ with $\operatorname{diam}(G[F][H]) > 2$. Then, there are two edges $e, f \in E(G[H])$ with $e, f \notin F$. It follows that $v_e, v_f \notin S$. But by construction of $\mathcal{I}'$, it holds that $\{v_e, v_f\} \in E(G')$, a contradiction to S being a vertex cover of G'. □

This concludes the proof of Proposition 3.26. Removing P_2s and P_3s with the linear-time Reduction Rules 3.5 and 3.7 yields the following corollary.

Corollary 3.34. 2-Diam GBP-C *can be solved in $\mathcal{O}(|V(G)|^2 \cdot \log |V(G)|)$ time on graphs of maximum degree at most four if each habitat has size at most three.*

3.2.3 NP-Hardness for $\Delta \geq 5$

In this section, we conduct the first of three proofs that follow a similar scheme. (With the other two proofs being the ones conducted in Sections 3.3.2 and 4.3.4.)

Proposition 3.35. 2-Diam GBP *is* NP*-hard even on graphs of maximum degree five with each habitat having size at most three.*

We give a polynomial-time reduction from the following NP-hard problem.

Problem: Cubic Vertex Cover
Input: A cubic graph G and an integer $k \in \mathbb{N}$.
Question: Is there a vertex cover $S \subseteq V(G)$ with $|S| \leq k$?

For easier notation, we denote an instance of the optimization version of 2-Diam GBP with graph G and habitat set $\mathcal{H}$ by $(G, \mathcal{H})$ without explicitly stating that we interpret $(G, \mathcal{H})$ as an instance of the optimization version of 2-Diam GBP and not, say, any other problem.

In the upcoming reduction, we replace each vertex of the given instance of Cubic Vertex Cover by a graph ("vertex gadget") and each edge by a graph ("edge gadget"). These gadgets mimic, in a sense, the nodes and edges of the graph of the Cubic Vertex Cover-instance. We define the gadgets in preparation for the reduction.

Definition 3.36. Let G be a graph and let $v \in V(G)$ be a vertex. The vertex gadget B_v corresponding to v is the graph with vertex set $V(B_v) := \{b_v^1, \ldots, b_v^5\}$ and edge set

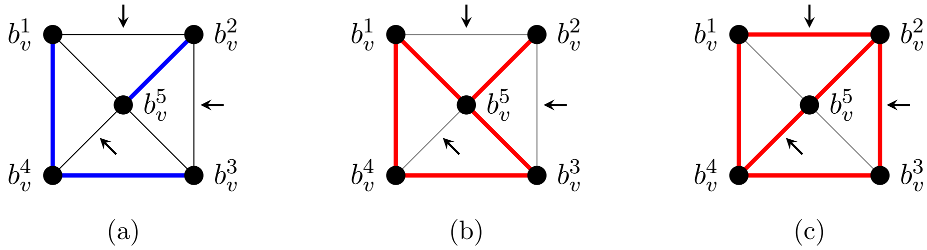


Figure 3.8: Illustration of vertex gadget B_v with habitat set $\mathcal{H}_v$ defined in Definition 3.36. Docking edges have arrows pointing to them. (a) Every solution contains all thick blue edges. (b) The thick red edges denote the unique solution of minimum size to $(B_v, \mathcal{H}_v)$. (c) The thick red edges denote a solution to $(B_v, \mathcal{H}_v)$ that is not minimum but includes all docking edges.

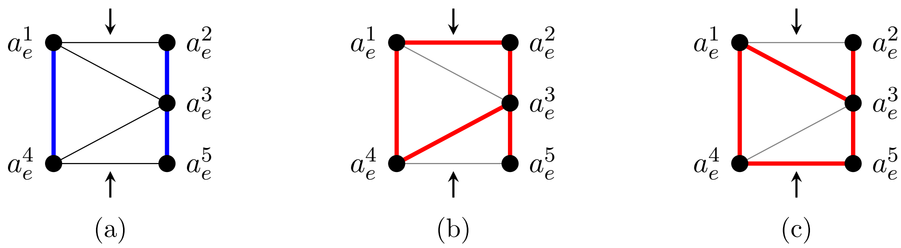


Figure 3.9: Illustration of edge gadget A_e with habitat set $\mathcal{H}_e$ defined in Definition 3.37. Docking edges have arrows pointing to them. (a) Every solution contains all thick blue edges. (b)&(c) In each subfigure the thick red edges denote a solution of minimum size to $(A_e, \mathcal{H}_e)$.

$E(B_v) := \{\{b_v^1, b_v^4\}\} \cup \bigcup_{i=1}^4 \{\{b_v^i, b_v^5\}\} \cup \bigcup_{i=1}^3 \{\{b_v^i, b_v^{i+1}\}\}$. We define the habitat set for B_v to be $\mathcal{H}_v := \{\{b_v^1, b_v^4\}, \{b_v^2, b_v^5\}, \{b_v^3, b_v^4\}\} \cup \{V(Q) \mid Q \subseteq B_v \text{ and } Q \text{ is a triangle}\}$. The docking edges of B_v are $\{b_v^1, b_v^2\}$, $\{b_v^2, b_v^3\}$, and $\{b_v^4, b_v^5\}$. If a vertex is an endvertex of a docking edge, then we call it a *docking vertex*. (See Figure 3.8a for an illustration.)

Docking edges are used to “glue together” vertex and edge gadgets in the reduction. Each instance $(B_v, \mathcal{H}_v)$ has the unique minimum solution shown in Figure 3.8b. The minimum solution does not contain any of the docking edges. However, as shown in Figure 3.8c, there is a solution that contains all docking edges and is only by one edge larger than the minimum solution.

Definition 3.37. Let G be a graph and let $e \in E(G)$ be an edge. The edge gadget A_e corresponding to e is the graph with vertex set $V(A_e) := \{a_e^1, \ldots, a_e^5\}$ and edge set $E(A_e) := E^* \cup \{\{a_e^1, a_e^2\}, \{a_e^1, a_e^3\}, \{a_e^3, a_e^4\}, \{a_e^4, a_e^5\}\}$ where the edge set E^* is defined as $E^* := \{\{a_e^1, a_e^4\}, \{a_e^2, a_e^3\}, \{a_e^3, a_e^5\}\}$. Moreover, the habitat set for A_e is defined as $\mathcal{H}_e := E^* \cup \{V(Q) \mid Q \subseteq A_e \text{ and } Q \text{ is a triangle}\}$. The docking edges of A_e are $\{a_e^1, a_e^2\}$ and $\{a_e^4, a_e^5\}$. (See Figure 3.9a for an illustration.)

Each instance $(A_e, \mathcal{H}_e)$ has multiple minimum solutions. For each docking edge there is a minimum solution containing it as shown in Figures 3.9b and 3.9c. There is no minimum solution containing both docking edges.

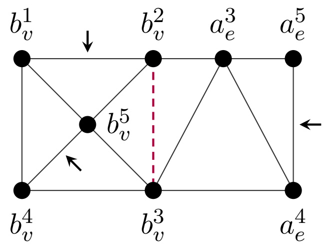


Figure 3.10: Result of gluing together a vertex gadget B_v and an edge gadget A_e using the docking edges $\{b_v^2, b_v^3\}$ and $\{a_e^1, a_e^2\}$. The used docking edge is dashed and purple. Unused docking edges have arrows pointing to them.

Gluing together a vertex gadget B_v and an edge gadget A_e means to select a previously unused docking edge $\{b_v^i, b_v^j\} \in E(B_v)$ and a previously unused docking edge $\{a_e^k, a_e^\ell\} \in E(A_e)$ and to set $a_e^k := b_v^i$ and $a_e^\ell := b_v^j$ (see Figure 3.10 for an illustration). To prevent vertex degrees from rising above five, we restrict ourselves to gluing together vertex and edge gadgets respecting the following rule. Whenever we glue a vertex gadget B_v and an edge gadget A_e together, we do it in such a way that the degree of $b_v^2 \in V(B_v)$ increases by at most one. We remark that we only ever glue together vertex and edge gadgets and never (directly) glue together a vertex gadget and a vertex gadget or an edge gadget and an edge gadget.

Observation 3.38. *Gluing together arbitrarily many vertex and edge gadgets as described above never results in a vertex having a degree larger than five.*

Proof. Let B_v be a vertex gadget that has been glued to three edge gadgets. Then, the degree of b_v^2 has increased twice, each time by at most one. Hence, b_v^2 has degree at most five. The degrees of the vertices b_v^1, b_v^3, b_v^4, and b_v^5 have each increased once. Since all docking vertices of edge gadgets have degree at most three, the degrees of the vertices b_v^1, b_v^3, b_v^4, and b_v^5 have each increased by at most two. Thus, the vertices b_v^1, b_v^3, b_v^4, and b_v^5 all have degree at most five. □

It is always possible to glue a vertex gadget and an edge gadget together at any given pair of docking edges such that the degree of $b_v^2 \in V(B_v)$ increases by at most one. This is because every previously unused docking edge of an edge gadget has a vertex of degree two. We describe the reduction next.

Construction 3.39. Let $\mathcal{I} = (G, k)$ be an instance of Cubic Vertex Cover with $V(G) = \{v_1, \ldots, v_n\}$, and $E(G) = \{e_1, \ldots, e_m\}$. Construct an instance $\mathcal{I}' = (G', \mathcal{H}', k)$ of 2-Diam GBP with $k' := 5n + 4m + k$ as follows.

Let G' and $\mathcal{H}'$ be initially empty. For each $v \in V(G)$ add the vertex gadget B_v to G' and extend $\mathcal{H}$ by $\mathcal{H}_v$. Then, follow the steps below for each $\{u, v\} \in E(G)$. Select a docking edge e_u of B_u and a docking edge e_v of B_v such that both docking edges have not been used before. Add A_e to G' such that A_e is glued to both B_u and B_v using the selected docking edges e_u and e_v. Moreover, extend $\mathcal{H}$ by $\mathcal{H}_e$. ◇

Observation 3.40. *The graph G' constructed in Construction 3.39 has maximum degree at most five.*

Observation 3.41. *The set $\mathcal{H}'$ constructed in Construction 3.39 only contains habitats of size at most three.*

Given a vertex gadget $B_v \subseteq G'$, we denote the set of docking edges of B_v by $E_{\text{dock}}(B_v)$. Likewise, given an edge gadget $A_e \subseteq G'$, we denote the set of docking edges of A_e by $E_{\text{dock}}(A_e)$. Furthermore, we use the notation $E_{\text{in}}(A_e) \coloneqq E(A_e) \setminus E_{\text{dock}}(A_e)$ to refer to the set of non-docking edges of A_e.

The following three observations are direct consequences of previously mentioned facts.

Observation 3.42. *Let $A_e \subseteq G'$ be an edge gadget and let $e' \in E_{\text{dock}}(A_e)$ be a docking edge of A_e. There is a subset $F_e \subseteq E_{\text{in}}(A_e)$ with $|F_e| \leq 4$ such that $F_e \cup \{e'\}$ is a solution to $(A_e, \mathcal{H}_e)$.*

Observation 3.43. *Let F be a solution to the constructed instance $\mathcal{I}'$ and let $B_v \subseteq G'$ be a vertex gadget. Then,*

$$|F \cap E(B_v)| \geq \begin{cases} 5, & \text{if } F \cap E_{\text{dock}}(B_v) = \emptyset, \\ 6, & \text{otherwise.} \end{cases}$$

Observation 3.44. *Let F be a solution to the constructed instance $\mathcal{I}'$ and let $A_e \subseteq G'$ be an edge gadget. Then,*

$$|F \cap E_{\text{in}}(A_e)| \geq \begin{cases} 5, & \text{if } F \cap E_{\text{dock}}(A_e) = \emptyset, \\ 4, & \text{otherwise.} \end{cases}$$

Lemma 3.45. *Let $\mathcal{I}'$ be the instance of* 2-Diam GBP *obtained from an instance $\mathcal{I}$ of* Cubic Vertex Cover. *Then, $\mathcal{I}$ is a* `yes`*-instance if and only if $\mathcal{I}'$ is a* `yes`*-instance.*

Proof. ($\Rightarrow$) Let $S \subseteq V(G)$ be a vertex cover of G with size at most k. Let $\bar{S} \coloneqq V(G) \setminus S$. The set

$$F_{\bar{S}} \coloneqq \bigcup_{v \in \bar{S}} \left(\{\{b_v^1, b_v^4\}, \{b_v^3, b_v^4\}\} \cup \bigcup_{i=1}^{3} \{\{b_v^i, b_v^5\}\} \right)$$

is a union of local solutions as depicted in Figure 3.8b. The set

$$F_S \coloneqq \bigcup_{v \in S} \{\{b_v^i, b_v^j\} \mid i \in \{1,3,5\} \wedge j \in \{2,4\}\}$$

is a union of local solutions as depicted in Figure 3.8c. Let $e \in E(G)$ be an edge. Since S is a vertex cover, at least one of the endvertices of e is contained in S. This implies that at least one of the docking edges of A_e is included in F_S. Thus, by Observation 3.42 there is a set F_e of size at most four such that $(F_e \cup F_S) \cap E(A_e)$ is a solution to $(A_e, \mathcal{H}_e)$. We claim that $F \coloneqq F_{\bar{S}} \cup F_S \cup \bigcup_{i=1}^{m} F_{e_j}$ is a solution to $\mathcal{I}'$. It holds that $|F| \leq 5 \cdot (n-k) + 6k + 4m = k'$. Since F is a union of local solutions, it holds that $\operatorname{diam}(G[F][H]) \leq 2$ for every $H \in \mathcal{H}$.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. Let $S' \coloneqq \{v \in V(G) \mid F \cap E_{\text{dock}}(B_v) \neq \emptyset\ \}$ and let S'' be a set constructed as follows. For each edge $e \in E(G)$ with $e \cap S' = \emptyset$ add one of the two endvertices of e to S''. Clearly, $S \coloneqq S' \cup S''$ is a vertex cover of G. It is left to show that $|S| \leq k$. Let $E_V \coloneqq \bigcup_{i=1}^{n} E(B_{v_i})$. By Observation 3.43 it holds that $|F \cap E_V| \geq 6 \cdot |S'| + 5 \cdot (n - |S'|) = 5n + |S'|$. Let $E_E \coloneqq \bigcup_{i=1}^{m} E_{\text{in}}(A_{e_i})$. Let ℓ be the number of edge gadgets $A_e \subseteq G'$ with the property that $F \cap E_{\text{dock}}(A_e) = \emptyset$. By Observation 3.44 it holds that $|F \cap E_E| \geq 5\ell + 4 \cdot (m - \ell) = 4m + \ell$. If for an edge $e \in E(G)$ it holds that $e \cap S' = \emptyset$, then $F \cap E_{\text{dock}}(A_e) = \emptyset$. By construction of S'', this implies that $\ell \geq |S''|$. It follows that $|F \cap E_E| \geq 4m + |S''|$. Hence, $|F| = |F \cap E_V| + |F \cap E_E| \geq 5n + |S'| + 4m + |S''|$. By rearrangement we get $|S| = |S'| + |S''| \leq |F| - 5n - 4m \leq k$. □

3.3 On Planar Graphs with Small Habitats

3.3.1 Polynomial-Time Solvable Cases

In this section, we show that 2-Diam GBP-C can be solved in polynomial time on planar graphs if each habitat has size at most three.

Herkenrath et al. [Her+22] show that 1-Reach GBP-C is solvable in polynomial time on plane graphs if each habitat induces a cycle which is the boundary of a face. If we restrict each habitat to induce a triangle, then 2-Diam GBP-C is equivalent to 1-Reach GBP-C by Lemma 3.74. Hence, 2-Diam GBP-C is solvable in polynomial time on plane graphs if each habitat induces a triangle which is the boundary of a face. We use this fact in the proof that 2-Diam GBP-C remains polynomial-time solvable if we drop the requirement that each induced triangle is the boundary of a face.

Proposition 3.46. 2-Diam GBP-C *can be solved in* $\mathcal{O}(|V(G)| \cdot |E(G)| \cdot |\mathcal{H}|^2 + |\mathcal{H}|^3)$ *time on plane graphs if each habitat induces a triangle.*

Let $\mathcal{A}_{\text{dec}}$ denote the polynomial-time algorithm by Herkenrath et al. [Her+22] for solving 2-Diam GBP-C on plane graphs with each habitat inducing a triangle which is the boundary of a face. It is not hard to deduce an algorithm $\mathcal{A}_{\text{opt}}$ that finds a minimum cost solution to an instance of 2-Diam GBP-C from the description of $\mathcal{A}_{\text{dec}}$ given by Herkenrath et al. [Her+22] if, again, the input graph is plane and each habitat induces a triangle which is the boundary of a face.

Let $(G, c, \mathcal{H}, k)$ be an instance of 2-Diam GBP-C where G is plane and each habitat induces a triangle.

The idea behind our algorithm is to iteratively simplify the given instance of 2-Diam GBP-C until it can be solved by algorithm $\mathcal{A}_{\text{dec}}$. Each step of simplification calls algorithm $\mathcal{A}_{\text{opt}}$ four times. More precisely, during a simplification step we find a habitat $H \in \mathcal{H}$ that separates the input graph such that one of the components obtained is easy to solve with the help of $\mathcal{A}_{\text{opt}}$. The information received by running $\mathcal{A}_{\text{opt}}$ enables us to delete some edges from $E(G)$ and some habitats from $\mathcal{H}$ if we carefully change k and the edge costs associated with the triangle $G[H]$. For an example of what the result of one step of simplification looks like, see Figure 3.11.

We will formulate the simplification step as a reduction rule. However, the reduction rule may change the cost function such that some edges in $E(G)$ are assigned non-positive cost. Thus, we modify 2-Diam GBP-C for the sake of this proof by allowing the cost function c to map to any value in $\mathbb{Z}$ instead of just $\mathbb{N}$. From the formulation given by

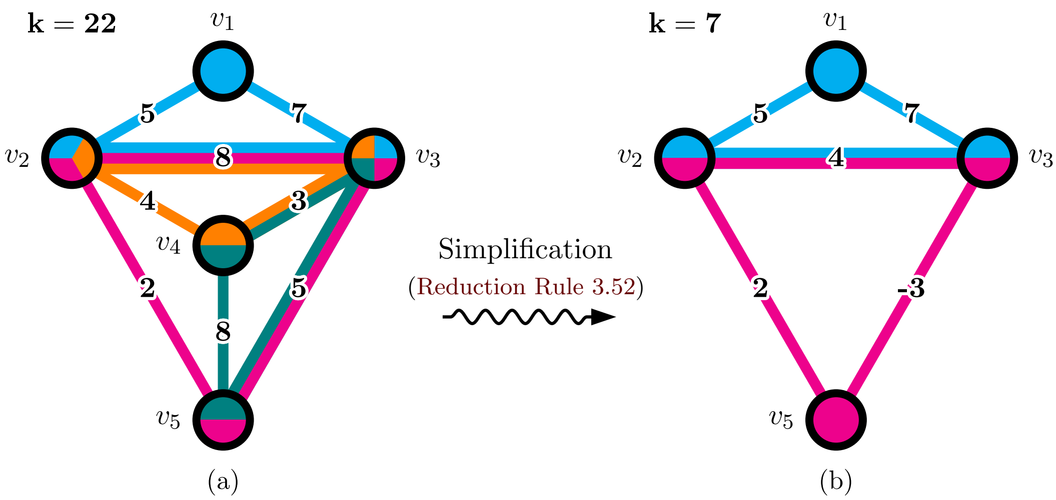


Figure 3.11: (a) An instance of 2-DIAM GBP-C where the graph is plane and each habitat induces a triangle. The triangles induced by habitats are marked by differing colors. Each edge has its cost written on it. (b) The instance of 2-DIAM GBP-C obtained after one step of simplification.

Herkenrath et al. [Her+22], it is not hard to see that $\mathcal{A}_{\text{dec}}$ and $\mathcal{A}_{\text{opt}}$ can be used even if c takes values in $\mathbb{Z}$. This is because the algorithms $\mathcal{A}_{\text{dec}}$ and $\mathcal{A}_{\text{opt}}$ are based on a reduction to MAXIMUM WEIGHT MATCHING.

Problem: MAXIMUM WEIGHT MATCHING
Input: An edge-weighted graph (G, w) and an integer $k \in \mathbb{N}$.
Task: Find a set $M \subseteq E(G)$ of maximum weight $w(M)$ such that for all $e, e' \in M$ it holds that $e \cap e' = \emptyset$.

Edges with non-positive weight can always be omitted from a maximum weight matching, so they can just be deleted before the actual matching algorithm is called. Moreover, we also allow k to take any value in $\mathbb{Z}$.

Throughout this proof we assume that every edge $e \in E(G)$ "is part of a habitat", i.e., that for every $e \in E(G)$ a habitat $H \in \mathcal{H}$ with $e \subseteq H$ exists. This can be achieved by exhaustively applying Reduction Rule 3.3 before doing anything else and then again exhaustively applying Reduction Rule 3.3 after each simplification step. Furthermore, we assume that G has no isolated vertices. This can similarly be achieved by exhaustively applying Reduction Rule 3.1 at the start and then again after each simplification step.

We make some definitions allowing us to relate the positions of objects to a habitat.

Definition 3.47. Let $H \in \mathcal{H}$ be a habitat, let $f_H^{\text{in}} \subseteq \mathbb{R}^2$ be the inner face of $G[H]$, and let $f_H^{\text{out}} \subseteq \mathbb{R}^2$ be the outer face of $G[H]$. We say that a vertex $v \in V(G)$ *lies within (outside of)* H if $v \in f_H^{\text{in}}$ ($v \in f_H^{\text{out}}$). We say that an edge $e \in E(G)$ *lies within (outside of)* H if at least one of its endvertices lies within (outside of) H. We say that a habitat $H' \in \mathcal{H}$ *lies within (outside of)* H if $f_{H'}^{\text{in}} \subsetneq f_H^{\text{in}}$ ($f_{H'}^{\text{out}} \subsetneq f_H^{\text{out}}$).

Since G is plane, for each habitat $H \in \mathcal{H}$ no edge $e \in E(G)$ lies both within H and outside of H. Likewise, no habitat $H' \in \mathcal{H}$ lies both within H and outside of H.

For a habitat $H \in \mathcal{H}$ we denote the set of edges lying within H by E_H^{in} and the set of edges lying outside of H by E_H^{out}. Moreover, we use the notation $E_H := E(G[H])$ to denote the edges of $G[H]$. Note that for every habitat $H \in \mathcal{H}$ it holds that $E(G) = E_H^{\text{in}} \uplus E_H \uplus E_H^{\text{out}}$. Furthermore, we denote the set of habitats lying within H by $\mathcal{H}_H^{\text{in}}$ and the set of habitats lying outside of H by $\mathcal{H}_H^{\text{out}}$. Note that for every habitat $H \in \mathcal{H}$ it holds that $\mathcal{H} = \mathcal{H}_H^{\text{in}} \uplus \{H\} \uplus \mathcal{H}_H^{\text{out}}$. Also note that for every habitat $H' \in \mathcal{H}_H^{\text{in}}$ it holds that $E(H') \subseteq E_H^{\text{in}} \cup E_H$ and for every habitat $H' \in \mathcal{H}_H^{\text{out}}$ it holds that $E(H') \in E_H^{\text{out}} \cup E_H$.

The following observation states that we can always find a habitat $H \in \mathcal{H}$ such that all habitats lying within H are "simple".

Observation 3.48. *Let $(G, c, \mathcal{H}, k)$ be an instance of* 2-Diam GBP-C *where G is plane, $\mathcal{H}$ is nonempty, and every habitat induces a triangle. There exists a habitat $H \in \mathcal{H}$ such that for every habitat $H' \in \mathcal{H}_H^{\text{in}}$ it holds that $\mathcal{H}_{H'}^{\text{in}} = \emptyset$.*

Proof. Assume towards a contradiction that no such habitat exists, i.e., for every habitat $H \in \mathcal{H}$ there is a habitat $H' \in \mathcal{H}_H^{\text{in}}$ with $\mathcal{H}_{H'}^{\text{in}} \neq \emptyset$. Hence, there exists an infinite sequence $(H)_{i \in \mathbb{N}}$ of habitats in $\mathcal{H}$ such that for every $i \in \mathbb{N}$ it holds that $H_{i+1} \in \mathcal{H}_{H_i}^{\text{in}}$. This means that for every $i \in \mathbb{N}$ the inner face f_{i+1} of $G[H_{i+1}]$ is a proper subset of the inner face f_i of $G[H_i]$. Since G is a (finite) graph, it holds that $V(G)$ has finite size. Consequently, the habitat set $\mathcal{H} \subseteq 2^{V(G)}$ also has finite size. It follows that there are numbers $i, j \in \mathbb{N}$ with $i < j$ such that $H_i = H_j$. Let f_i be the inner face of $G[H_i]$ and let f_j be the inner face of $G[H_j]$. By transitivity of the $\subsetneq$-relation on sets, we get $f_j \subsetneq f_i$, a contradiction to $H_i = H_j$. □

Observation 3.48 permits the case that for every habitat $H \in \mathcal{H}$ it holds that $\mathcal{H}_H^{\text{in}} = \emptyset$. But in this case we can apply $\mathcal{A}_{\text{dec}}$ to solve the instance of 2-Diam GBP-C. Furthermore, if $\mathcal{H}$ is empty, then a solution is easily constructed by choosing the set of all edges with negative cost as the solution. Thus, we can assume that a habitat defined as follows exists.

Definition 3.49. We call a habitat $H \in \mathcal{H}$ a *reduction base habitat* if $\mathcal{H}_H^{\text{in}} \neq \emptyset$ and for all $H' \in \mathcal{H}_H^{\text{in}}$ it holds that $\mathcal{H}_{H'}^{\text{in}} = \emptyset$.

For an example of a reduction base habitat see the habitat $\{v_2, v_3, v_5\}$ (colored pink) in Figure 3.11a. Note that a reduction base habitat $H \in \mathcal{H}$ separates the set of vertices within H and the set of vertices outside of H. Next, we define two types of graphs. We use these graphs as input graphs for $\mathcal{A}_{\text{opt}}$.

Definition 3.50. Let $H \in \mathcal{H}$ be a habitat. The *basic inner graph* (G_H, c_H) of H is an edge-weighted graph with $G_H := G[E_H^{\text{in}} \cup E_H]$ and for all $e \in E(G_H)$ it holds that

$$c_H(e) := \begin{cases} c(e), & \text{if } e \in E_H^{\text{in}}, \\ 0, & \text{otherwise.} \end{cases}$$

For an example of a basic inner graph, see Figure 3.12a. For a habitat $H \in \mathcal{H}$ we denote by F_H a set obtained as follows. Let (G_H, c_H) be the basic inner graph of H and let F' be a solution to the instance $(G_H, c_H, \mathcal{H}_H^{\text{in}})$ of (the optimization version of) 2-Diam GBP-C of minimum cost. Then, $F_H := F' \cap E_H^{\text{in}}$. Note that F' (and thereby F_H) can be computed by applying $\mathcal{A}_{\text{opt}}$ if H is a reduction base habitat.

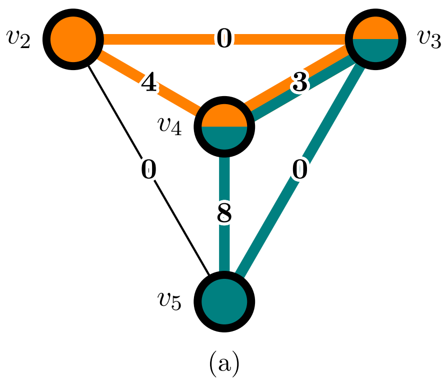


Figure 3.12: (a) The basic inner graph of habitat $H := \{v_2, v_3, v_5\}$ with regard to the instance of 2-Diam GBP-C depicted in Figure 3.11a. Additionally, the triangles induced by the habitats lying within H are marked by differing colors. (b) The $\{v_2, v_3\}$-omitting inner graph of H.

Definition 3.51. Let $H \in \mathcal{H}$ be a habitat and let $x \in E_H$. The *x-omitting inner graph* (G_H, c_H) of H is an edge-weighted graph with $G_H := G[E_H^{\text{in}} \cup E_H]$ and for all $e \in E(G_H)$ it holds that

$$c_H(e) := \begin{cases} c(e), & \text{if } e \in E_H^{\text{in}}, \\ c(\{e \in E_H^{\text{in}} \mid c(e) > 0\}) + 1, & \text{if } e = x, \\ 0, & \text{otherwise.} \end{cases}$$

For an example of a x-omitting inner graph, see Figure 3.12b. For a habitat $H \in \mathcal{H}$ and an edge $x \in E_H$ we denote by F_H^x a set obtained as follows. Let (G_H, c_H) be the x-omitting inner graph of H and let F' be a solution to the instance $(G_H, c_H, \mathcal{H}_H^{\text{in}})$ of (the optimization version of) 2-Diam GBP-C of minimum cost. Then, $F_H^x := F' \cap E_H^{\text{in}}$. Note that F' (and thereby F_H^x) can be computed by applying $\mathcal{A}_{\text{opt}}$ if H is a reduction base habitat.

With the notation ready, we can formulate the reduction rule.

Reduction Rule 3.52. *Select a reduction base habitat $H \in \mathcal{H}$. Set $k := k + 2 \cdot c(F_H) - \sum_{e \in E_H} c(F_H^e)$ and for each $e \in E_H$ set $c(e) := c(e) + c(F_H) - c(F_H^e)$. Delete all habitats in $\mathcal{H}_H^{\text{in}}$ from $\mathcal{H}$. Delete all edges in E_H^{in} from $E(G)$. Finally, restrict the cost function c to $E(G)$.*

A single application of Reduction Rule 3.52 requires calling algorithm $\mathcal{A}_{\text{opt}}$ four times. One call is made to compute F_H and the other three calls are made to compute F_H^e for every $e \in E_H$. Before proving correctness of the reduction rule, we make three observations. The first of these observations is immediate.

Observation 3.53. *An edge set $F \subseteq E(G)$ is a solution if and only if $c(F) \leq k$ and for each habitat $H \in \mathcal{H}$ it holds that two or three edges from E_H are contained in F.*

Observation 3.54. *Let F be a solution and let $H \in \mathcal{H}$ be a habitat. It holds that $c(F_H) \leq c(F \cap E_H^{\text{in}})$.*

Proof. Let (G_H, c_H) be the basic inner graph of H. Remember that F_H is obtained by finding a minimum cost solution F' to the instance $\mathcal{I} := (G_H, c_H, \mathcal{H}_H^{\text{in}})$ of 2-Diam GBP-C and setting $F_H := F' \cap E_H^{\text{in}}$. Since $c_H(e) = 0$ for every $e \in E_H$, it holds that $c_H(F') = c_H(F' \cap E_H^{\text{in}}) = c_H(F_H)$. Because F' is a solution of minimum cost to $\mathcal{I}$ and because $F \cap E_H^{\text{in}}$ is a solution (of indeterminate cost) to $\mathcal{I}$, it follows that $c_H(F') \leq c_H(F \cap E_H^{\text{in}})$. Thus, $c_H(F_H) \leq c_H(F \cap E_H^{\text{in}})$. Since $c_H(e) = c(e)$ for every $e \in E_H^{\text{in}}$, it follows that $c(F_H) \leq c(F \cap E_H^{\text{in}})$. □

Observation 3.55. *Let F be a solution and let $H \in \mathcal{H}$ be a habitat such that there is a unique edge $x \in E_H$ with $x \notin F$. It holds that $c(F_H^x) \leq c(F \cap E_H^{\text{in}})$.*

Proof. Let (G_H, c_H) be the x-omitting inner graph of H. Remember that F_H^x is obtained by finding a minimum cost solution F' to the instance $\mathcal{I} := (G_H, c_H, \mathcal{H}_H^{\text{in}})$ of 2-Diam GBP-C and setting $F_H^x := F' \cap E_H^{\text{in}}$. Let $y, z \in E_H$ be the two distinct edges with $y, z \in F$. We show that $x \notin F'$. Assume towards a contradiction that $x \in F'$. Then, it holds that

$$\begin{aligned}
c_H(F') &= c_H(x) + c_H(F' \setminus \{x\}) \\
&= c(\{e \in E_H^{\text{in}} \mid c(e) > 0\}) + 1 + c_H(F' \setminus \{x\}) \\
&> c_H(E_H^{\text{in}}) \\
&= c_H(E_H^{\text{in}} \cup \{y, z\}).
\end{aligned}$$

But this means that $E_H^{\text{in}} \cup \{y, z\}$ is a solution to $\mathcal{I}$ of lower cost than F', a contradiction. Thus, because of $x \notin F'$ and $c_H(y) = c_H(z) = 0$, it follows that $c_H(F') = c_H(F' \cap E_H^{\text{in}}) = c(F_H^x)$. Because F' is a solution of minimum cost to $\mathcal{I}$ and because $F \cap E_H^{\text{in}}$ is a solution (of indeterminate cost) to $\mathcal{I}$, it follows that $c_H(F') \leq c_H(F \cap E_H^{\text{in}})$. Thus, $c_H(F_H^x) \leq c_H(F \cap E_H^{\text{in}})$. Since $c_H(e) = c(e)$ for every $e \in E_H^{\text{in}}$, it follows that $c(F_H^x) \leq c(F \cap E_H^{\text{in}})$. □

Lemma 3.56. *Let $\mathcal{I}' = (G', c', \mathcal{H}', k')$ be the instance of* 2-Diam GBP-C *obtained from an instance $\mathcal{I} = (G, c, \mathcal{H}, k)$ of* 2-Diam GBP-C *by application of Reduction Rule 3.52. Then, $\mathcal{I}$ is a* `yes`*-instance if and only if $\mathcal{I}'$ is a* `yes`*-instance.*

Proof. In the following, let the notations E_H^{in}, E_H, and E_H^{out} refer to instance $\mathcal{I}$. But note that $E_H \subseteq E(G')$ and $E_H^{\text{out}} \subseteq E(G')$.

(⇒) Let F be a solution to $\mathcal{I}$. Let H be the reduction base habitat selected in the application of Reduction Rule 3.52. We claim that $F' := F \setminus E_H^{\text{in}}$ is a solution to $\mathcal{I}'$. We will make use of the fact that the following holds.

$$\begin{aligned}
c'(F') &= c'(F \setminus E_H^{\text{in}}) \\
&= c'(F \cap (E_H^{\text{out}} \cup E_H)) \\
&= c(F \cap E_H^{\text{out}}) + c'(F \cap E_H) \\
&= c(F) - c(F \cap E_H^{\text{in}}) - c(F \cap E_H) + c'(F \cap E_H) \\
&\leq k - c(F \cap E_H^{\text{in}}) - c(F \cap E_H) + c'(F \cap E_H)
\end{aligned}$$

We show that $c'(F') \leq k'$. For this we distinguish two cases, one of which holds because of Observation 3.53.

First case: All three edges from E_H are contained in F. This results in the following.

$$\begin{aligned}
c'(F') &\leq k - c(F \cap E_H^{\text{in}}) - c(F \cap E_H) + c'(F \cap E_H) \\
&\leq k - c(F_H) - c(F \cap E_H) + c'(F \cap E_H) && \text{(Obs. 3.54)} \\
&= k - c(F_H) - \sum_{e \in E_H} c(e) + \sum_{e \in E_H} \Big(c(e) + c(F_H) - c(F_H^e) \Big) \\
&= k + 2 \cdot c(F_H) - \sum_{e \in E_H} c(F_H^e) = k'
\end{aligned}$$

Second case: Only two edges from E_H are contained in F. Let $x \in E_H$ be the unique edge that is not included in F and let $y, z \in E_H$ be the two distinct edges that are included in F. This results in the following.

$$\begin{aligned}
c'(F') &\leq k - c(F \cap E_H^{\text{in}}) - c(F \cap E_H) + c'(F \cap E_H) \\
&\leq k - c(F_H^x) - c(F \cap E_H) + c'(F \cap E_H) && \text{(Obs. 3.55)} \\
&= k - c(F_H^x) - \sum_{e \in \{y,z\}} c(e) + \sum_{e \in \{y,z\}} \Big(c(e) + c(F_H) - c(F_H^e) \Big) \\
&= k + 2 \cdot c(F_H) - \sum_{e \in E_H} c(F_H^e) = k'
\end{aligned}$$

This concludes the case distinction. From F being a solution to $\mathcal{I}$ and Observation 3.53, it follows that two or three edges from $E_{H'}$ are contained in F for every $H' \in \mathcal{H}$. Because of the way $\mathcal{H}'$ is constructed by Reduction Rule 3.52, it holds that $E_{H'} \subseteq E(G) \setminus E_H^{\text{in}}$ for every $H' \in \mathcal{H}'$. This means that for every $H' \in \mathcal{H}'$ two or three edges from $E_{H'}$ are contained in $F \setminus E_H^{\text{in}} = F'$. Thus, F' is a solution to $\mathcal{I}'$ by Observation 3.53.

($\Leftarrow$) Let H be the reduction base habitat selected in the application of Reduction Rule 3.52 and let $e \in E_H$. We will make use of the following reformulations of equations.

$$\begin{aligned}
c'(e) = c(e) + c(F_H) - c(F_H^e) &\iff c(e) = c'(e) - c(F_H) + c(F_H^e) \\
k' = k + 2 \cdot c(F_H) - \sum_{e \in E_H} c(F_H^e) &\iff k = k' - 2 \cdot c(F_H) + \sum_{e \in E_H} c(F_H^e)
\end{aligned}$$

Let F' be a solution to $\mathcal{I}'$. Due to Observation 3.53 one of the following two cases holds.

First case: All three edges from E_H are contained in F'. We claim that $F := F' \cup F_H$

is a solution to $\mathcal{I}$. We show that $c(F) \leq k$.

$$\begin{aligned}
c(F) &= c(F' \cup F_H) \\
&= c(F') + c(F_H) \\
&= c'(F' \cap E_H^{\text{out}}) + c(F' \cap E_H) + c(F_H) \\
&= c'(F' \cap E_H^{\text{out}}) + \sum_{e \in E_H} \Big(c'(e) - c(F_H) + c(F_H^e) \Big) + c(F_H) \\
&= c'(F' \cap E_H^{\text{out}}) + \sum_{e \in E_H} c'(e) - 2 \cdot c(F_H) + \sum_{e \in E_H} c(F_H^e) \\
&= c'(F') - 2 \cdot c(F_H) + \sum_{e \in E_H} c(F_H^e) \\
&\leq k' - 2 \cdot c(F_H) + \sum_{e \in E_H} c(F_H^e) = k
\end{aligned}$$

From F' being a solution to $\mathcal{I}'$ and Observation 3.53, it follows that two or three edges from $E_{H'}$ are contained in F' for every $H' \in \{H\} \cup \mathcal{H}_H^{\text{out}}$. From the definition of F_H and Observation 3.53, it follows that two or three edges from $E_{H'}$ are contained in F_H for every $H' \in \mathcal{H}_H^{\text{in}}$. Thus, for every $H' \in \mathcal{H}$ two or three edges from E_H are contained in $F' \cup F_H = F$. Consequently, F is a solution to $\mathcal{I}$ by Observation 3.53.

Second case: Only two edges from E_H are contained in F'. Let $x \in E_H$ be the unique edge that is not included in F' and let $y, z \in E_H$ be the two distinct edges that are included in F'. We claim that $F := F' \cup F_H^x$ is a solution to $\mathcal{I}$. We show that $c(F) \leq k$.

$$\begin{aligned}
c(F) &= c(F' \cup F_H^x) \\
&= c(F') + c(F_H^x) \\
&= c'(F' \cap E_H^{\text{out}}) + c(F' \cap E_H) + c(F_H^x) \\
&= c'(F' \cap E_H^{\text{out}}) + \sum_{e \in \{y,z\}} \Big(c'(e) - c(F_H) + c(F_H^e) \Big) + c(F_H^x) \\
&= c'(F' \cap E_H^{\text{out}}) + \sum_{e \in \{y,z\}} c'(e) - 2 \cdot c(F_H) + \sum_{e \in E_H} c(F_H^e) \\
&= c'(F') - 2 \cdot c(F_H) + \sum_{e \in E_H} c(F_H^e) \\
&\leq k' - 2 \cdot c(F_H) + \sum_{e \in E_H} c(F_H^e) = k
\end{aligned}$$

Analogously to the first case, from F' being a solution to $\mathcal{I}'$ and Observation 3.53, it follows that two or three edges from $E_{H'}$ are contained in F' for every $H' \in \{H\} \cup \mathcal{H}_H^{\text{out}}$. From the definition of F_H^x and Observation 3.53, it follows that two or three edges from $E_{H'}$ are contained in F_H^x for every $H' \in \mathcal{H}_H^{\text{in}}$. Thus, for every $H' \in \mathcal{H}$ two or three edges from E_H are contained in $F' \cup F_H^x = F$. Consequently, F is a solution to $\mathcal{I}$ by Observation 3.53. $\square$

For a single habitat $H \in \mathcal{H}$ it is possible to compute $\mathcal{H}_H^{\text{in}}$ in $\mathcal{O}(|\mathcal{H}|)$ time by iterating through all habitats $H' \in \mathcal{H}$ and checking whether H' lies within H. Thus, computing $\mathcal{H}_H^{\text{in}}$ for every $H \in \mathcal{H}$ can be done in $\mathcal{O}(|\mathcal{H}|^2)$ time. Having computed $\mathcal{H}_H^{\text{in}}$ for every $H \in \mathcal{H}$, finding a reduction base habitat or detecting that no reduction base habitat exists in $\mathcal{H}$ is doable in $\mathcal{O}(|\mathcal{H}|^2)$ time. Each habitat in $\mathcal{H}$ is selected as the reduction base habitat H in Reduction Rule 3.52 at most once. This means that Reduction Rule 3.52 is applied at most $\mathcal{O}(|\mathcal{H}|)$ times. Hence, a reduction base habitat needs to be searched for at most $\mathcal{O}(|\mathcal{H}|)$ times. The total time spent for finding reduction base habitats thereby is in $\mathcal{O}(|\mathcal{H}|^3)$.

Since Reduction Rule 3.52 is applied at most $\mathcal{O}(|\mathcal{H}|)$ times, algorithm $\mathcal{A}_{\text{opt}}$ is applied at most $\mathcal{O}(|\mathcal{H}|)$ times. Algorithm $\mathcal{A}_{\text{dec}}$ is only applied once at the end. Since a single application of $\mathcal{A}_{\text{opt}}$ and $\mathcal{A}_{\text{dec}}$ as described by Herkenrath et al. [Her+22] has a running time of $\mathcal{O}(|V(G)| \cdot |E(G)| \cdot |\mathcal{H}|)$, the overall time needed for running $\mathcal{A}_{\text{opt}}$ and $\mathcal{A}_{\text{dec}}$ is in $\mathcal{O}(|V(G)| \cdot |E(G)| \cdot |\mathcal{H}|^2)$. All other actions performed by Reduction Rule 3.52 take only linear time.

The Reduction Rules 3.1 and 3.3 are exhaustively applied once more often than Reduction Rule 3.52 is singly applied. However, the exhaustive application of both Reduction Rules 3.1 and 3.3 is asymptotically faster than a single application of Reduction Rule 3.52. Consequently, we do not need to further consider the time necessary for running Reduction Rules 3.1 and 3.3. Thus, the resulting total running time is $\mathcal{O}(|V(G)| \cdot |E(G)| \cdot |\mathcal{H}|^2 + |\mathcal{H}|^3)$. This concludes the proof of Proposition 3.46.

Removing P_2s and P_3s with the linear-time Reduction Rules 3.5 and 3.7 yields the following corollary.

Corollary 3.57. 2-Diam GBP-C *can be solved in* $\mathcal{O}(|V(G)| \cdot |E(G)| \cdot |\mathcal{H}|^2 + |\mathcal{H}|^3)$ *time on planar graphs if each habitat has size at most three.*

3.3.2 NP-Hardness

In this section, we show the following.

Proposition 3.58. 2-Diam GBP *is* NP-*hard even on planar graphs of maximum degree five with each habitat having size at most four.*

The following problem is NP-hard [Moh01].

Problem: Planar Cubic Vertex Cover
Input: A planar cubic graph G and an integer $k \in \mathbb{N}$.
Question: Is there a vertex cover $S \subseteq V(G)$ with $|S| \leq k$?

We give a polynomial-time reduction from Planar Cubic Vertex Cover.

For easier notation, we denote an instance of the optimization version of 2-Diam GBP with graph G and habitat set $\mathcal{H}$ by $(G, \mathcal{H})$ without explicitly stating that we interpret $(G, \mathcal{H})$ as an instance of the optimization version of 2-Diam GBP and not, say, any other problem.

In the upcoming reduction, we replace each vertex of the given instance of Planar Cubic Vertex Cover by a graph ("vertex gadget") and each edge by a graph ("edge gadget"). These gadgets mimic, in a sense, the nodes and edges of the graph of the

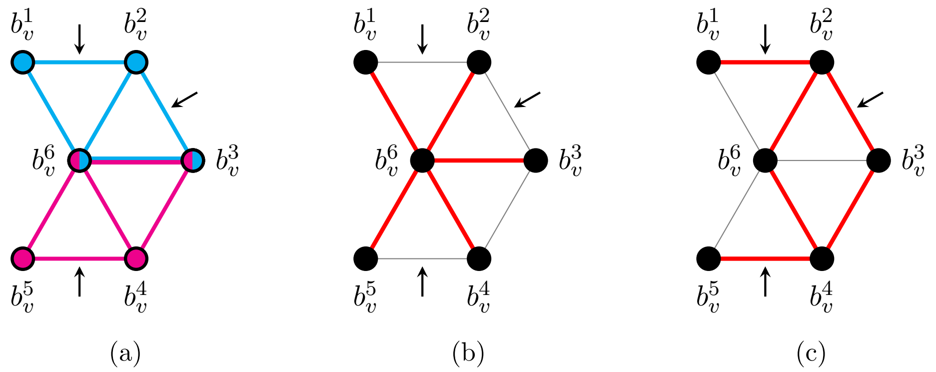


Figure 3.13: Illustration of vertex gadget B_v with habitat set $\mathcal{H}_v$ defined in Definition 3.59. Docking edges have arrows pointing to them. (a) The two subgraphs induced by the two habitats in $\mathcal{H}_v$ are marked by two different colors. (b) The thick red edges denote the unique solution of minimum size to $(B_v, \mathcal{H}_v)$. (c) The thick red edges denote a solution to $(B_v, \mathcal{H}_v)$ that is not minimum but includes all docking edges.

Planar Cubic Vertex Cover-instance. We define the gadgets in preparation for the reduction.

Definition 3.59. Let G be a graph and let $v \in V(G)$ be a vertex. The vertex gadget B_v corresponding to v is the graph with vertex set $V(B_v) := \{b_v^1, \dots, b_v^6\}$ and edge set $E(B_v) := \bigcup_{i=1}^{5}\{\{b_v^i, b_v^6\}\} \cup \bigcup_{i=1}^{4}\{\{b_v^i, b_v^{i+1}\}\}$. We define the habitat set for B_v to be $\mathcal{H}_v := \{\{b_v^1, b_v^2, b_v^3, b_v^6\}, \{b_v^3, b_v^4, b_v^5, b_v^6\}\}$. The *docking edges* of B_v are $\{b_v^1, b_v^2\}$, $\{b_v^2, b_v^3\}$, and $\{b_v^4, b_v^5\}$. If a vertex is an endvertex of a docking edge, then we call it a *docking vertex*. (See Figure 3.13a for an illustration.)

Docking edges are used to "glue together" vertex and edge gadgets in the reduction. (The choice of the docking edges is somewhat arbitrary. Because of symmetry, we could use $\{b_v^3, b_v^4\}$ instead of $\{b_v^2, b_v^3\}$ as a docking edge.)

Each instance $(B_v, \mathcal{H}_v)$ has the unique minimum solution shown in Figure 3.13b. The minimum solution does not contain any of the docking edges. However, as shown in Figure 3.13c, there is a solution that contains all docking edges and is only by one edge larger than the minimum solution.

Note that the degrees of docking vertices vary. Furthermore, the vertex b_v^2 is a docking vertex shared by two docking edges. These properties of docking vertices are relevant when deciding how to glue together vertex and edge gadgets. The decisions must be made carefully so that no vertex degree increases above five and the resulting graph is planar.

We define not just one but two edge gadgets. In the reduction each edge of the instance of Planar Cubic Vertex Cover is replaced by just one edge gadget. But it might be necessary to replace some edges by the first edge gadget and other edges by the second edge gadget.

Definition 3.60. Let G be a graph and let $e \in E(G)$ be an edge. The *straight edge gadget* A_e^{str} corresponding to e is the graph with vertex set $V(A_e^{\text{str}}) := \{a_e^1, \dots, a_e^7\}$ and

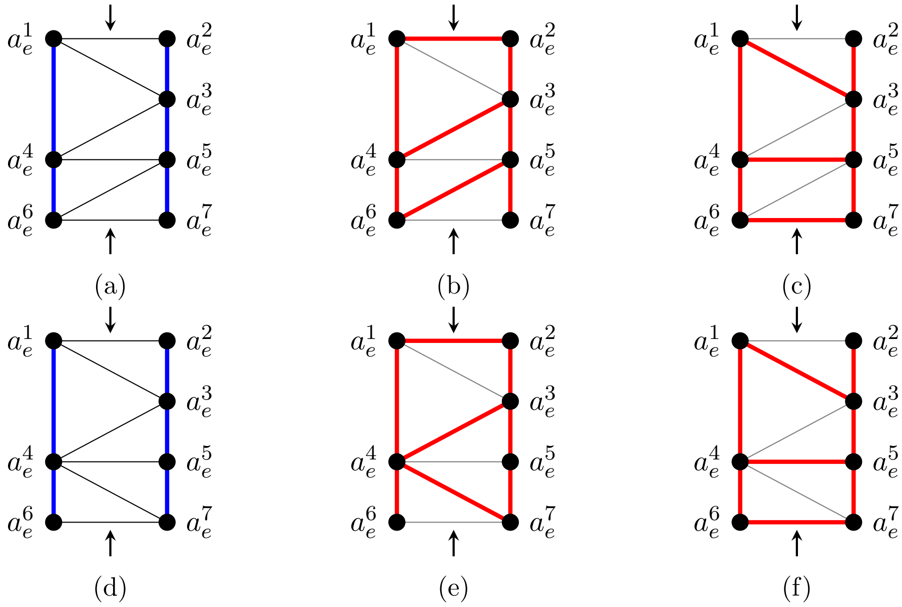


Figure 3.14: (a)–(c) Illustration of edge gadget A_e^{str} with habitat set $\mathcal{H}_e^{\mathrm{str}}$ defined in Definition 3.60. (d)–(f) Illustration of edge gadget A_e^{inv} with habitat set $\mathcal{H}_e^{\mathrm{inv}}$ defined in Definition 3.61. (a)–(f) Docking edges have arrows pointing to them. (a)&(d) Every solution contains all thick blue edges. (b)&(c) In each subfigure the thick red edges denote a solution of minimum size to $(A_e^{\mathrm{str}}, \mathcal{H}_e^{\mathrm{str}})$. (e)&(f) In each subfigure the thick red edges denote a solution of minimum size to $(A_e^{\mathrm{inv}}, \mathcal{H}_e^{\mathrm{inv}})$.

edge set $E(A_e^{\mathrm{str}}) := E_e^* \cup \{\{a_e^1, a_e^2\}, \{a_e^1, a_e^3\}, \{a_e^3, a_e^4\}, \{a_e^4, a_e^5\}, \{a_e^5, a_e^6\}, \{a_e^6, a_e^7\}\}$ where $E_e^* := \{\{a_e^1, a_e^4\}, \{a_e^4, a_e^6\}, \{a_e^2, a_e^3\}, \{a_e^3, a_e^5\}, \{a_e^5, a_e^7\}\}$. The habitat set for A_e^{str} is $\mathcal{H}_e^{\mathrm{str}} := E_e^* \cup \{V(Q) \mid Q \subseteq A_e^{\mathrm{str}}$ and Q is a triangle$\}$. The *docking edges* of A_e^{str} are $\{a_e^1, a_e^2\}$ and $\{a_e^6, a_e^7\}$. (See Figure 3.14a for an illustration.)

Each instance $(A_e^{\mathrm{str}}, \mathcal{H}_e^{\mathrm{str}})$ has multiple minimum solutions. For each docking edge there is a minimum solution containing it as shown in Figures 3.14b and 3.14c. There is no minimum solution containing both docking edges. Again, note that the degrees of docking vertices vary.

Definition 3.61. Let G be a graph and let $e \in E(G)$ be an edge. The *degree-inverting edge gadget* A_e^{inv} corresponding to e is obtained by letting A_e^{inv} be initially equal to A_e^{str} and then deleting the edge $\{a_e^5, a_e^6\}$ and adding the edge $\{a_e^4, a_e^7\}$. The habitat set for A_e^{inv} is $\mathcal{H}_e^{\mathrm{inv}} := E_e^* \cup \{V(Q) \mid Q \subseteq A_e^{\mathrm{inv}}$ and Q is a triangle$\}$. The *docking edges* of A_e^{inv} are $\{a_e^1, a_e^2\}$ and $\{a_e^6, a_e^7\}$. (See Figure 3.14d for an illustration.)

As for straight edge gadgets, each instance $(A_e^{\mathrm{inv}}, \mathcal{H}_e^{\mathrm{inv}})$ has multiple minimum solutions. For each docking edge there is a minimum solution containing it as shown in Figures 3.14e and 3.14f. There is no minimum solution containing both docking edges. Note that the degrees of docking vertices vary.

Gluing together a vertex gadget B_v and an edge gadget A_e means to select a previously unused docking edge $\{b_v^i, b_v^j\} \in E(B_v)$ and a previously unused docking edge

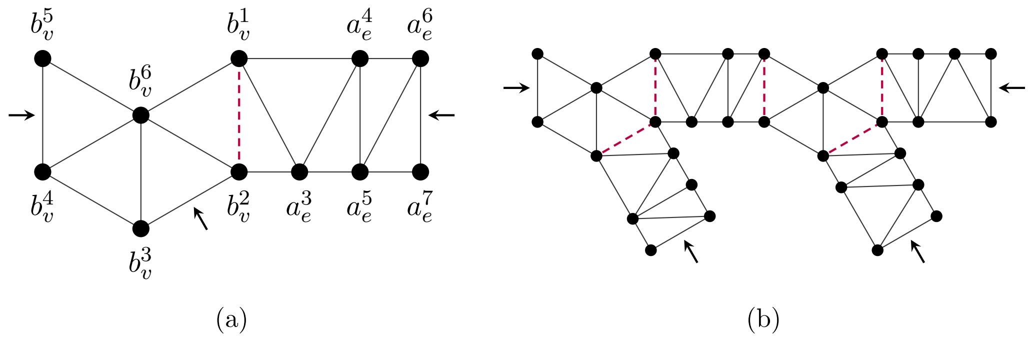


Figure 3.15: Used docking edges are dashed and purple. Unused docking edges have arrows pointing to them. (a) Result of gluing together a vertex gadget B_v and a straight edge gadget A_e^{str} using the docking edges $\{b_v^1, b_v^2\}$ and $\{a_e^1, a_e^2\}$. (b) Result of gluing together multiple vertex and edge gadgets.

$\{a_e^k, a_e^\ell\} \in E(A_e)$ and to set $a_e^k := b_v^i$ and $a_e^\ell := b_v^j$ (see Figure 3.15 for an illustration). To prevent vertex degrees from rising above five, we restrict ourselves to gluing together vertex and edge gadgets respecting the following rule. Whenever we glue a vertex gadget B_v and an edge gadget A_e together, we do it in such a way that the degree of $b_v^2 \in V(B_v)$ increases by at most one. We remark that we only ever glue together vertex and edge gadgets and never (directly) glue together a vertex gadget and a vertex gadget or an edge gadget and an edge gadget.

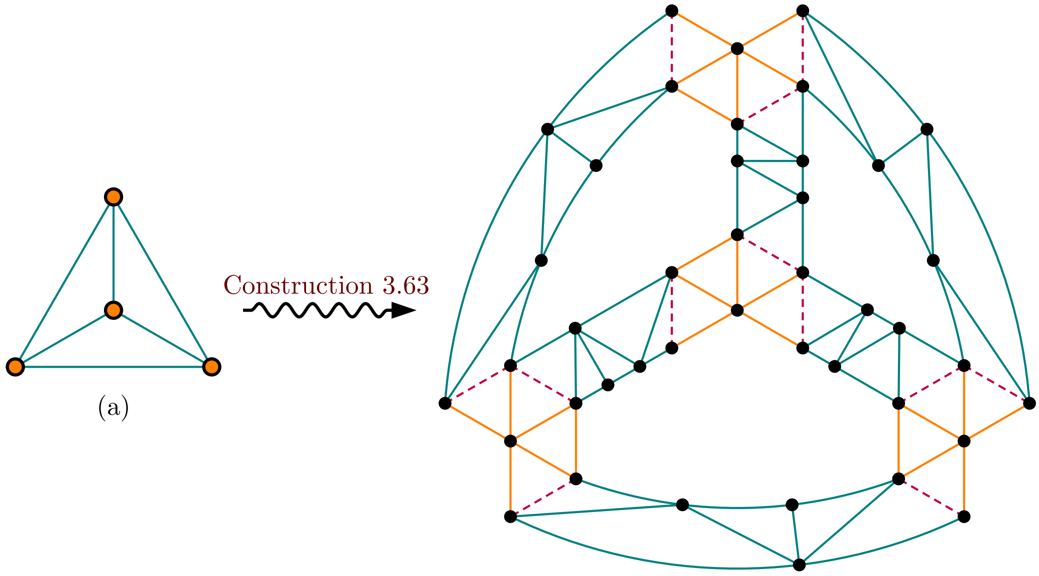
Observation 3.62. *Gluing together arbitrarily many vertex and edge gadgets as described above never results in a vertex having a degree larger than five.*

Proof. Let B_v be a vertex gadget that has been glued to three edge gadgets. Then, the degree of b_v^2 has increased twice, each time by at most one. Hence, b_v^2 has degree at most five. The degrees of the vertices b_v^1, b_v^3, b_v^4, and b_v^5 have each increased once. Since all docking vertices of edge gadgets have degree at most three, the degrees of the vertices b_v^1, b_v^3, b_v^4, and b_v^5 have each increased by at most two. Thus, the vertices b_v^1, b_v^3, b_v^4, and b_v^5 all have degree at most five. □

It is always possible to glue a vertex gadget and an edge gadget together at any given pair of docking edges such that the degree of $b_v^2 \in V(B_v)$ increases by at most one. This is because every previously unused docking edge of an edge gadget has a vertex of degree two.

In the upcoming reduction, when we replace an edge from the graph G of the instance of Planar Cubic Vertex Cover by an edge gadget, having defined two types of edge gadgets will allow us to choose one of the edge gadgets. This ability to choose will enable us to maintain planarity of the graph G' of the constructed instance of 2-Diam GBP while meeting the condition that for each $v \in V(G)$ the degree of $b_v^2 \in V(B_v)$ increases by at most one when gluing together a vertex and an edge gadget.

The reason that having two edge gadgets helps in maintaining planarity can be seen by comparing A_e^{str} and A_e^{inv} in Figures 3.14a and 3.14d. In Figure 3.14a the docking vertices having degree two (i.e., a_e^2 and a_e^7) are both on one side (on the right), whereas



(a) (b)

Figure 3.16: Illustration to Construction 3.63. (a) Example graph G of the input instance $\mathcal{I} = (G, k)$. Vertices are orange and edges are turquoise. (b) A graph G' that can be obtained from G using Construction 3.63. Docking edges (being both part of a vertex and an edge gadget) are dashed and purple. Edges that are only part of a vertex gadget are orange. Edges that are only part of an edge gadget are turquoise.

in Figure 3.14d the docking vertices having degree two are on different sides (a_e^2 on the right and a_e^6 on the left). Using this we can ensure that the docking vertices of degree two are always placed on the side they need to be on.

Construction 3.63. Let $\mathcal{I} = (G, k)$ be an instance of Planar Cubic Vertex Cover with $V(G) = \{v_1, \ldots, v_n\}$, and $E(G) = \{e_1, \ldots, e_m\}$. Construct an instance $\mathcal{I}' = (G', \mathcal{H}', k)$ of 2-Diam GBP with $k' := 5n + 7m + k$ as follows (see Figure 3.16 for an illustration).

We create a plane drawing D' of G' by gradually making additions to D'. Whenever we make an addition to D', we assume that we make the additions in a reasonable manner using the insights gained prior to this construction. Let D' be initially a drawing of the empty graph and let $\mathcal{H}'$ be initially empty. For each $v \in V(G)$ add a drawing of the vertex gadget B_v to D' and extend $\mathcal{H}$ by $\mathcal{H}_v$. Then, follow the steps below for each $\{u, v\} \in E(G)$. Select a docking edge e_u of B_u and a docking edge e_v of B_v such that both docking edges have not been used before. Next, add a drawing of A_e^{str} or A_e^{inv} to D' such that it is glued to both B_u and B_v using the selected docking edges e_u and e_v. Moreover, extend $\mathcal{H}$ by the corresponding set $\mathcal{H}_e^{\text{str}}$ or $\mathcal{H}_e^{\text{inv}}$. Finally, let G' be the graph corresponding to the drawing D'. $\diamond$

Observation 3.64. *The graph G' constructed in Construction 3.63 is planar and has maximum degree at most five.*

Observation 3.65. *The set $\mathcal{H}'$ constructed in Construction 3.63 only contains habitats of size at most four.*

Since the constructed graph G' contains only one edge gadget for every edge $e \in E(G)$ of the original graph G, we can simply denote this edge gadget by A_e instead of A_e^{str} or A_e^{inv}. Likewise, we can write $\mathcal{H}_e$ for the associated habitat set instead of $\mathcal{H}_e^{\mathrm{str}}$ or $\mathcal{H}_e^{\mathrm{inv}}$. Given a vertex gadget $B_v \subseteq G'$, we denote the set of docking edges of B_v by $E_{\mathrm{dock}}(B_v)$. Likewise, given an edge gadget $A_e \subseteq G'$, we denote the set of docking edges of A_e by $E_{\mathrm{dock}}(A_e)$. Furthermore, we use the notation $E_{\mathrm{in}}(A_e) := E(A_e) \setminus E_{\mathrm{dock}}(A_e)$ to refer to the set of non-docking edges of A_e.

The following three observations are direct consequences of previously mentioned facts.

Observation 3.66. *Let $A_e \subseteq G'$ be an edge gadget and let $e' \in E_{\mathrm{dock}}(A_e)$ be a docking edge of A_e. There is a subset $F_e \subseteq E_{\mathrm{in}}(A_e)$ with $|F_e| \leq 7$ such that $F_e \cup \{e'\}$ is a solution to $(A_e, \mathcal{H}_e)$.*

Observation 3.67. *Let F be a solution to the constructed instance $\mathcal{I}'$ and let $B_v \subseteq G'$ be a vertex gadget. Then,*

$$|F \cap E(B_v)| \geq \begin{cases} 5, & \textit{if } F \cap E_{\mathrm{dock}}(B_v) = \emptyset, \\ 6, & \textit{otherwise.} \end{cases}$$

Observation 3.68. *Let F be a solution to the constructed instance $\mathcal{I}'$ and let $A_e \subseteq G'$ be an edge gadget. Then,*

$$|F \cap E_{\mathrm{in}}(A_e)| \geq \begin{cases} 8, & \textit{if } F \cap E_{\mathrm{dock}}(A_e) = \emptyset, \\ 7, & \textit{otherwise.} \end{cases}$$

Lemma 3.69. *Let $\mathcal{I}'$ be the instance of* 2-Diam GBP *obtained from an instance $\mathcal{I}$ of* Planar Cubic Vertex Cover*. Then, $\mathcal{I}$ is a* `yes`*-instance if and only if $\mathcal{I}'$ is a* `yes`*-instance.*

Proof. ($\Rightarrow$) Let $S \subseteq V(G)$ be a vertex cover of G with size at most k. Let $\bar{S} := V(G) \setminus S$. The set

$$F_{\bar{S}} := \bigcup_{v \in \bar{S}} \left(\bigcup_{i=1}^{5} \{\{b_v^i, b_v^6\}\} \right)$$

is a union of local solutions as depicted in Figure 3.13b. The set

$$F_S := \bigcup_{v \in S} \left(\{\{b_v^2, b_v^6\}, \{b_v^4, b_v^6\}\} \cup \bigcup_{i=1}^{4} \{\{b_v^i, b_v^{i+1}\}\} \right)$$

is a union of local solutions as depicted in Figure 3.13c. Let $e \in E(G)$ be an edge. Since S is a vertex cover, at least one of the endvertices of e is contained in S. This implies that at least one of the docking edges of A_e is included in F_S. Thus, by Observation 3.66 there is a set F_e of size at most seven such that $(F_e \cup F_S) \cap E(A_e)$ is a solution to $(A_e, \mathcal{H}_e)$. We claim that $F := F_{\bar{S}} \cup F_S \cup \bigcup_{i=1}^{m} F_{e_j}$ is a solution to $\mathcal{I}'$. It holds that

$|F| \leq 5 \cdot (n-k) + 6k + 7m = k'$. Since F is a union of local solutions, it holds that $\operatorname{diam}(G[F][H]) \leq 2$ for every $H \in \mathcal{H}$.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. Let $S' := \{v \in V(G) \mid F \cap E_{\text{dock}}(B_v) \neq \emptyset\ \}$ and let S'' be a set constructed as follows. For each edge $e \in E(G)$ with $e \cap S' = \emptyset$ add one of the two endvertices of e to S''. Clearly, $S := S' \cup S''$ is a vertex cover of G. It is left to show that $|S| \leq k$. Let $E_V := \bigcup_{i=1}^{n} E(B_{v_i})$. By Observation 3.67 it holds that $|F \cap E_V| \geq 6 \cdot |S'| + 5 \cdot (n - |S'|) = 5n + |S'|$. Let $E_E := \bigcup_{i=1}^{m} E_{\text{in}}(A_{e_i})$. Let ℓ be the number of edge gadgets $A_e \subseteq G'$ with the property that $F \cap E_{\text{dock}}(A_e) = \emptyset$. By Observation 3.68 it holds that $|F \cap E_E| \geq 8\ell + 7 \cdot (m - \ell) = 7m + \ell$. If for an edge $e \in E(G)$ it holds that $e \cap S' = \emptyset$, then $F \cap E_{\text{dock}}(A_e) = \emptyset$. By construction of S'', this implies that $\ell \geq |S''|$. It follows that $|F \cap E_E| \geq 7m + |S''|$. Hence, $|F| = |F \cap E_V| + |F \cap E_E| \geq 5n + |S'| + 7m + |S''|$. By rearrangement we get $|S| = |S'| + |S''| \leq |F| - 5n - 7m \leq k$. □

3.4 Structural Parameterizations

3.4.1 Number of Triangles

We show that 2-Diam GBP-C is fixed-parameter tractable with respect to the number of triangles contained in the input graph. Moreover, we identify some polynomial-time solvable cases of 2-Diam GBP-C in a corollary.

Let $\mathcal{I} = (G, \mathcal{H}, c, k)$ be an instance of 2-Diam GBP-C and let $\#C_3$ denote the number of triangles contained in G.

Proposition 3.70. 2-Diam GBP-C *can be solved in* $8^{\#C_3}|\mathcal{I}|^{\mathcal{O}(1)}$ *time.*

We recall two reduction rules.

Reduction Rule 3.3 (Restated)**.** *If for an edge $e \in E(G)$ there is no habitat $H \in \mathcal{H}$ with $e \subseteq H$, then delete e.*

Reduction Rule 3.9 (Restated)**.** *If for an edge $e \in E(G)$ there is a habitat $H \in \mathcal{H}$ with $e \subseteq H$ such that e is not contained in a triangle in G, then fix $e \in F$.*

After the exhaustive application of the above two reduction rules, every edge of the obtained graph is contained in a triangle or fixed to be included in the solution F. Thus, it remains to decide for at most $3 \cdot \#C_3$ edges which ones to add to the solution F. This can be done by a brute-force approach, systematically checking all $2^{3 \cdot \#C_3}$ possibilities. This concludes the proof of Proposition 3.70.

Remark. 2-Diam GBP-C can be solved in $2^t|\mathcal{I}|^{\mathcal{O}(1)}$ time, where t is the number of edges contained in triangles.

Since, e.g., bipartite graphs contain no odd cycles and therefore no triangles, we obtain the following.

Corollary 3.71. 2-Diam GBP-C *can be solved in polynomial time*

- *on bipartite graphs.*
- *on graphs of treewidth at most one.*
- *on graphs with vertex cover number at most one.*

3.4.2 Feedback Edge Number

In this section, we show that 2-Diam GBP-C is fixed-parameter tractable with respect to the feedback edge number of the input graph. As a part of the process, we prove a lemma that we also use later in the context of $(2,2)$-Closed GBP-C. (Later usage is the reason why we are interested in cycles of length at most six.)

Lemma 3.72. *Let G be a graph. The number $\#C_{\leq 6}$ of cycles of length at most six contained in G is upper bounded by a function only depending on the feedback edge number of G.*

Proof. Let $F \subseteq E(G)$ be a feedback edge set of G. Let $n_F := \sum_{k=0}^{6} \binom{|F|}{k}$. We claim that $\#C_{\leq 6} \leq n_F$. It holds that each cycle of length at most six in G contains at least one edge from F. Assume towards a contradiction that there are more than n_F cycles of length at most six contained in G. Then, by pigeonhole principle, there is a subset $F' \subseteq F$ such that there are two distinct cycles $C_1, C_2 \subseteq G$ with $E(C_1) \cap F = E(C_2) \cap F = F'$. Let $E_{\text{diff}} := E(C_1) \triangle E(C_2)$ and let $G_{\text{diff}} := G[E_{\text{diff}}]$. Since C_1 and C_2 are distinct, it holds that $E_{\text{diff}} \neq \emptyset$. Due to $E(C_1) \cap F = E(C_2) \cap F$, it holds that $E_{\text{diff}} \cap F = \emptyset$. Thus, $G_{\text{diff}} \subseteq G - F$. This implies that G_{diff} is a forest. Therefore, there is a vertex $v \in V(G_{\text{diff}})$ with $\deg_{G_{\text{diff}}}(v) = 1$. Let $e_v^{\text{diff}} \in E_{\text{diff}}$ be the unique edge in E_{diff} with $v \in e_v^{\text{diff}}$. Assume w.l.o.g. that $e_v^{\text{diff}} \in E(C_1)$ (otherwise interchange the names of C_1 and C_2). Since C_1 is a cycle, there is a unique edge $e_v \in E(C_1)$ with $v \in e_v$ and $e_v \neq e_v^{\text{diff}}$. Because of $\deg_{G_{\text{diff}}}(v) = 1$, it holds that $e_v \notin E_{\text{diff}}$. Hence, $e_v \in E(C_2)$. Since C_2 is a cycle, there is a unique edge $e'_v \in E(C_2)$ with $v \in e'_v$ and $e'_v \neq e_v$. Because of $e_v^{\text{diff}} \in E(C_1)$, it holds that $e'_v \neq e_v^{\text{diff}}$. As e_v^{diff} is the only edge in E_{diff} with $v \in e_v^{\text{diff}}$, it follows that $e'_v \notin E_{\text{diff}}$. This implies that $e'_v \in E(C_1)$. In summary, the edges $e_v^{\text{diff}}, e_v, e'_v$ are included in $E(C_1)$, pairwise distinct, and all have v as an endvertex. Thus, $\deg_{C_1}(v) \geq 3$, a contradiction to C_1 being a cycle. □

We have seen in the previous section that 2-Diam GBP-C is fixed-parameter tractable with respect to the number of triangles contained in the input graph (see Proposition 3.70). Using the above lemma, we get the following.

Proposition 3.73. 2-Diam GBP-C *is fixed-parameter tractable with respect to the feedback edge number of the input graph.*

3.4.3 Distance to Clique

We formally prove that 2-Diam GBP and 1-Reach GBP are equivalent when restricted to instances where habitats have size at most three. Subsequently, we use a result by Herkenrath et al. [Her+22] showing NP-hardness of 2-Diam GBP on cliques.

For the following equivalence proof, note that every instance of 2-Diam GBP is also an instance of 1-Reach GBP and vice versa.

Lemma 3.74. *Let $\mathcal{I} = (G, \mathcal{H}, k)$ be an instance of* 2-Diam GBP *with habitats of size at most three. Then, $\mathcal{I}$ is a* `yes`*-instance of* 2-Diam GBP *if and only if $\mathcal{I}$ is a* `yes`*-instance of* 1-Reach GBP.

Proof. (⇒) Let F be a solution to $\mathcal{I}$ interpreted as an instance of 2-Diam GBP. This means that for every habitat $H \in \mathcal{H}$ it holds that $\operatorname{diam}(G[F][H]) \leq 2$. Thus, for every habitat $H \in \mathcal{H}$ the graph $G[F][H]$ is connected. It follows that F is a solution to $\mathcal{I}$ interpreted as an instance of 1-Reach GBP.

(⇐) Similarly, let F be a solution to $\mathcal{I}$ interpreted as an instance of 1-Reach GBP. This means that for every habitat $H \in \mathcal{H}$ the graph $G[F][H]$ is connected. Since every habitat has size at most three, this implies that for every habitat $H \in \mathcal{H}$ it holds that $\operatorname{diam}(G[F][H]) \leq 2$. Thus, F is a solution to $\mathcal{I}$ interpreted as an instance of 2-Diam GBP. □

Herkenrath et al. [Her+22] show that 1-Reach GBP is NP-hard even if G is a clique and each habitat induces a P_2 or a C_3. Therefore, the following holds.

Proposition 3.75. 2-Diam GBP *is* NP-*hard even if G is a clique and each habitat induces a P_2 or a C_3.*

3.4.4 Vertex Cover Number

We show that 2-Diam GBP is NP-hard even on graphs with vertex cover number three, on graphs of treewidth three, and on graphs with distance to bipartite one. Additionally, we exclude the existence of polynomial kernels for 2-Diam GBP regarding some parameterizations. We start by considering the vertex cover number.

Proposition 3.76. 2-Diam GBP *is* NP-*hard even on graphs with vertex cover number three.*

The following proof is inspired by the proof that 1-Reach GBP is NP-hard even on series-parallel graphs given by Fluschnik and Kellerhals [FK21]. In Sections 4.3.1 and 4.3.2, we conduct two more proofs of this kind. We give a polynomial-time reduction from the following NP-hard problem.

Problem: Hitting Set
Input: A universe U, a set $\mathcal{F} \subseteq 2^U$, and an integer $k \in \mathbb{N}$.
Question: Is there a subset $U' \subseteq U$ with $|U'| \leq k$ such that for every $F \in \mathcal{F}$ it holds that $F \cap U' \neq \emptyset$?

Construction 3.77. Let $\mathcal{I} = (U, \mathcal{F}, k)$ be an instance of Hitting Set with $U = \{1, \ldots, n\}$ and $\mathcal{F} = \{F_1, \ldots, F_m\}$. Construct an instance $\mathcal{I}' = (G, \mathcal{H}, k')$ of 2-Diam GBP with $k' := 2n + k + 1$ as follows (see Figure 3.17 for an illustration).

Let G be initially empty. Add the vertex set $V_U := \{x_i \mid i \in U\}$ and the three vertices s, t, and h. Moreover, add the edge sets $E^* := \{\{t, h\}\} \cup \bigcup_{i=1}^{n} \{\{s, x_i\}, \{x_i, h\}\}$ and $E_U := \bigcup_{i=1}^{n} \{\{x_i, t\}\}$. Finally, let $\mathcal{H} := E^* \cup \{Z_1, \ldots, Z_m\}$ with $Z_j := \{s, t, h\} \cup \bigcup_{i \in F_j} \{x_i\}$ for every $F_j \in \mathcal{F}$. ⋄

We make two observations.

Observation 3.78. *The graph G constructed in Construction 3.77 has vertex cover number three.*

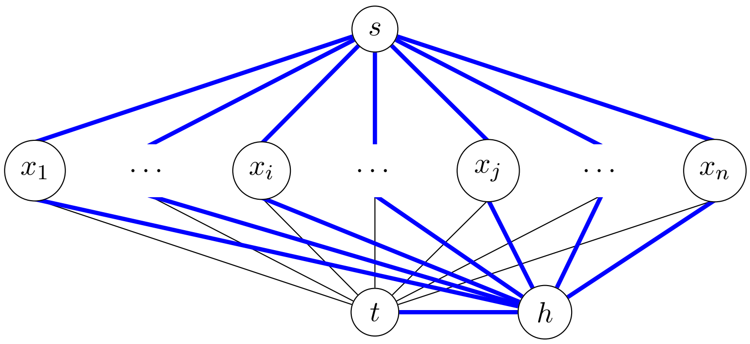


Figure 3.17: Illustration to Construction 3.77. Every solution contains all thick blue edges.

Observation 3.79. *Let $\mathcal{I}'$ be a* yes*-instance obtained by Construction 3.77. Then, every solution F contains all edges in E^*.*

By the following lemma, we prove Proposition 3.76.

Lemma 3.80. *Let $\mathcal{I}'$ be the instance of* 2-Diam GBP *obtained from an instance $\mathcal{I}$ of* Hitting Set *using Construction 3.77. Then, $\mathcal{I}$ is a* yes*-instance if and only if $\mathcal{I}'$ is a* yes*-instance.*

Proof. ($\Rightarrow$) Let $U' \subseteq U$ be a solution to $\mathcal{I}$. We claim that $F := E^* \cup \bigcup_{i \in U'} \{\{x_i, t\}\}$ is a solution to $\mathcal{I}'$. Note that $|F| \leq 2n+k+1$. Since $E^* \subseteq F$, it holds that $\operatorname{diam}(G[F][H]) \leq 2$ for every habitat $H \in E^*$. Assume towards a contradiction that there is a habitat $Z_j \in \mathcal{H}$ such that $\operatorname{diam}(G[F][Z_j]) > 2$. From $E^* \subseteq F$ it follows that any two vertices $u, v \in Z_j$ with $\{u, v\} \neq \{s, t\}$ have a distance of at most two in $G[F][Z_j]$. Hence, $\operatorname{dist}_{G[F][Z_j]}(s, t) > 2$. This has the consequence that no edge from $\{\{x_i, t\} \mid i \in F_j\}$ is included in F. Therefore, $F_j \cap U' = \emptyset$, a contradiction to U' being a solution to $\mathcal{I}$.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. We claim that $U' := \{i \in U \mid \{x_i, t\} \in F\}$ is a solution to $\mathcal{I}$. Note that it follows from $E^* \subseteq F$ that $|U'| \leq k$. Assume towards a contradiction that U' is not a solution. Then, there is an $F_j \in \mathcal{F}$ with $F_j \cap U' = \emptyset$. Hence, no edge from the edge set $E_j := \{\{x_i, t\} \mid i \in F_j\}$ is included in F. Since every (s, t)-path of length at most two in $G[Z_j]$ contains an edge from E_j, it follows that $\operatorname{dist}_{G[F][Z_j]}(s, t) > 2$. Thus, it holds that $\operatorname{diam}(G[F][Z_j]) > 2$, a contradiction to F being a solution to $\mathcal{I}'$. $\square$

We make two further observations regarding the graph constructed in Construction 3.77.

Observation 3.81. *The graph G constructed in Construction 3.77 has treewidth three.*

Observation 3.82. *The graph G constructed in Construction 3.77 has distance to bipartite one.*

These observations yield the following corollary.

Corollary 3.83. 2-Diam GBP *is* NP*-hard*

- *even on graphs of treewidth three.*

- *on graphs with distance to bipartite one.*

Finally, we discuss polynomial kernels. Hitting Set parameterized by the size of the universe U does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly [DLS14]. Since the graph G' constructed in Construction 3.77 only has $|U| + 3$ vertices, virtually every natural graph parameter of G' is polynomially bounded by $|U|$. Thus, Construction 3.77 is a polynomial parameter transformation from Hitting Set parameterized by $|U|$ to 2-Diam GBP parameterized by virtually any natural graph parameter. As a consequence, we get the following.

Proposition 3.84. *Unless* NP $\subseteq$ coNP/poly, 2-Diam GBP *parameterized by virtually any natural graph parameter admits no polynomial kernel.*

Chapter 4

$(2,2)$-Closed GBP

4.1 Preprocessing

In this section, we define and analyze reduction rules for later usage. We consider an instance $\mathcal{I} = (G, c, \mathcal{H}, k)$ of $(2,2)$-Closed GBP-C.

Definition 4.1. For a habitat $H \in \mathcal{H}$, we define the 2-*habitat* H^2 corresponding to H as $H^2 := H \cup \{x \in V(G) \mid \exists y, z \in H.\, y \neq z \wedge x \in N_G(y) \cap N_G(z)\}$. We define the set of 2-habitats $\mathcal{H}^2$ corresponding to $\mathcal{H}$ as $\mathcal{H}^2 := \{H^2 \mid H \in \mathcal{H}\}$.

Intuitively, a 2-habitat H^2 is the set of vertices the animals of the species modeled by habitat $H \in \mathcal{H}$ are able to move through in the context of $(2,2)$-Closed GBP-C. The following is an adaption of Reduction Rule 3.3 to $(2,2)$-Closed GBP-C.

Reduction Rule 4.2. *If for an edge $e \in E(G)$ there is no 2-habitat $H^2 \in \mathcal{H}^2$ with $e \subseteq H^2$, then delete e.*

Observation 4.3. *Reduction Rule 4.2 is correct.*

Proof. Let $\mathcal{I}' = (G', c', \mathcal{H}', k')$ be the instance of $(2,2)$-Closed GBP-C obtained from $\mathcal{I}$ by application of Reduction Rule 4.2. We show that $\mathcal{I}$ is a `yes`-instance if and only if $\mathcal{I}'$ is a `yes`-instance. Let $e \in E(G)$ be the edge deleted in the application of Reduction Rule 4.2.

($\Rightarrow$) Let F be a solution to $\mathcal{I}$. We claim that $F' := F \setminus \{e\}$ is a solution to $\mathcal{I}'$. Clearly, $c(F') \leq k'$. Assume towards a contradiction that there is a habitat $H \in \mathcal{H}$ with vertices $u, v \in H$ such that $\operatorname{dist}_{G[F']^2[H]}(u, v) > 2$. Since $\operatorname{dist}_{G[F]^2[H]}(u, v) \leq 2$, it follows that there is a (u, v)-path of length at most two in $G[H]$ such that $e \in E(P)$. But this implies that $e \subseteq H^2$, a contradiction.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. Then, F is also a solution to $\mathcal{I}$. $\square$

The following is an adaption of Reduction Rule 3.9 to $(2,2)$-Closed GBP-C. It also adds an edge to the solution F that is being constructed.

Reduction Rule 4.4. *If for an edge $e \in E(G)$ there is a 2-habitat $H^2 \in \mathcal{H}^2$ with $e \subseteq H^2$ such that e is not contained in a cycle of length at most six in G, then fix $e \in F$.*

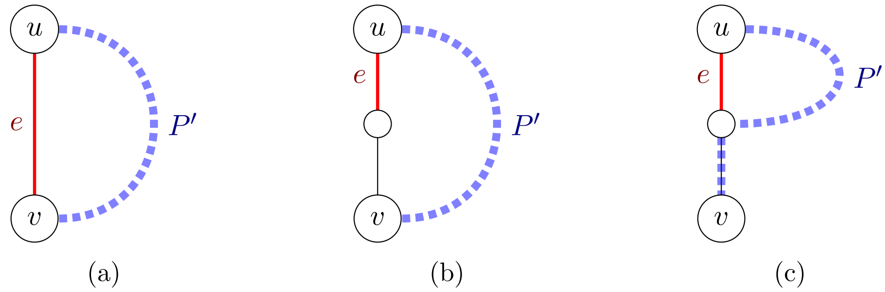


Figure 4.1: Illustration to the proof of Observation 4.5. The edge e has been fixed to be included in F in the application of Reduction Rule 4.4. The (u, v)-path P' has length at most four. Each way the vertices u, v, the edge e, and the path P' can relate to each other (up to symmetry by interchanging u and v) is depicted in one subfigure. In each subfigure the edge e is contained in a cycle of length at most six.

Observation 4.5. *Reduction Rule 4.4 is correct.*

Proof. Let $e \in E(G)$ be the edge fixed to be included in F in the application of Reduction Rule 4.4. Assume towards a contradiction that there is a solution $\widetilde{F}$ to $\mathcal{I}$ with $e \notin \widetilde{F}$. Let $H^2 \in \mathcal{H}^2$ be a 2-habitat for which $e \subseteq H^2$ holds. Let $H \in \mathcal{H}$ be the habitat corresponding to H^2. Then, there are vertices $u, v \in H$ and a (u, v)-path P of length at most two in G such that $e \in E(P)$. By definition of (2, 2)-Closed GBP-C, it holds that $\operatorname{dist}_{G[\widetilde{F}]^2}(u, v) \leq 2$. By definition of the second power of a graph, it follows that $\operatorname{dist}_{G[\widetilde{F}]}(u, v) \leq 4$. Hence, there is a (u, v)-path P' of length at most four in G such that $e \notin E(P')$. The existence of P and P' implies that e is contained in a cycle of length at most six in G (see Figure 4.1). This is a contradiction. □

4.2 Tractability

4.2.1 On Trees and Cycles

In this section, we show the following.

Proposition 4.6. (2, 2)-Closed GBP-C *can be solved in polynomial time on graphs where each component is a tree or a cycle.*

Moreover, at the end of this section, we record some more specialized cases in a corollary. We consider an instance $\mathcal{I} = (G, \mathcal{H}, c, k)$ of (2, 2)-Closed GBP-C where each component of G is a tree or a cycle. We recall two reduction rules.

Reduction Rule 4.2 (Restated)**.** *If for an edge $e \in E(G)$ there is no 2-habitat $H^2 \in \mathcal{H}^2$ with $e \subseteq H^2$, then delete e.*

Reduction Rule 4.4 (Restated)**.** *If for an edge $e \in E(G)$ there is a 2-habitat $H^2 \in \mathcal{H}^2$ with $e \subseteq H^2$ such that e is not contained in a cycle of length at most six in G, then fix $e \in F$.*

Exhaustive application of the above reduction rules results in every component of G being an isolated vertex or a cycle of length at most six. If there is a habitat $H \in \mathcal{H}$ with vertices $u, v \in H$ such that u and v are contained in different components of G, then $\operatorname{diam}(G) = \infty$, which means that `no` is to be returned. If no such habitat H exists, then for each component C of G, we can independently search for a minimum cost local solution $F_C \subseteq E(C)$ satisfying $\operatorname{diam}(C[F_C][H_C]) \leq 2$ for every $H_C \in \mathcal{H}$ with $H_C \subseteq V(C)$. Depending on the success of finding these local solutions and their summed cost, `yes` or `no` is to be returned.

Corollary 4.7. $(2,2)$-Closed GBP-C *can be solved in polynomial time*

- *on graphs of maximum degree at most two.*
- *on graphs of treewidth at most one.*
- *on graphs with vertex cover number at most one.*

4.2.2 Parameterized by Feedback Edge Number

In this section, we show that $(2,2)$-Closed GBP-C is fixed-parameter tractable with respect to the feedback edge number f of the input graph. The proof is similar to the proof that 2-Diam GBP-C is fixed-parameter tractable with respect to f (see Proposition 3.73). Let $\mathcal{I} = (G, \mathcal{H}, c, k)$ be an instance of $(2,2)$-Closed GBP-C and let $\#C_{\leq 6}$ denote the number of cycles of length at most six contained in G. We start by proving the following proposition.

Proposition 4.8. $(2,2)$-Closed GBP-C *can be solved in* $64^{\#C_{\leq 6}}|\mathcal{I}|^{\mathcal{O}(1)}$ *time.*

We recall two reduction rules.

Reduction Rule 4.2 (Restated)**.** *If for an edge $e \in E(G)$ there is no 2-habitat $H^2 \in \mathcal{H}^2$ with $e \subseteq H^2$, then delete e.*

Reduction Rule 4.4 (Restated)**.** *If for an edge $e \in E(G)$ there is a 2-habitat $H^2 \in \mathcal{H}^2$ with $e \subseteq H^2$ such that e is not contained in a cycle of length at most six in G, then fix $e \in F$.*

After the exhaustive application of the above two reduction rules, every edge of the obtained graph is contained in a cycle of length at most six or fixed to be included in the solution F. Thus, it remains to decide for at most $6 \cdot \#C_{\leq 6}$ edges which ones to add to the solution F. This can be done by a brute-force approach, systematically checking all $2^{6 \cdot \#C_{\leq 6}}$ possibilities. This concludes the proof of Proposition 4.8.

Remark. 2-Diam GBP-C can be solved in $2^t|\mathcal{I}|^{\mathcal{O}(1)}$ time, where t is the number of edges contained in cycles of length at most six.

Using Lemma 3.72 we get the following.

Proposition 4.9. $(2,2)$-Closed GBP-C *is fixed-parameter tractable with respect to the feedback edge number of the input graph.*

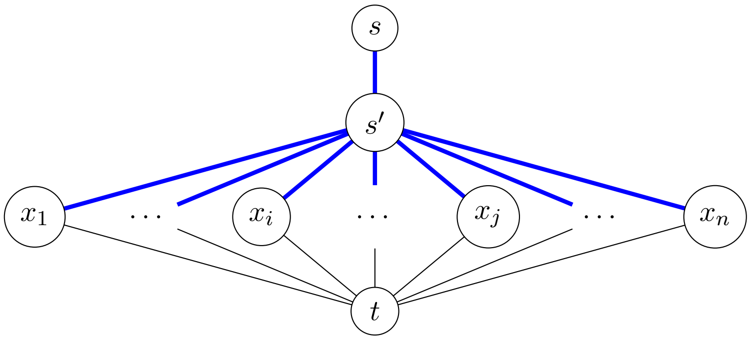


Figure 4.2: Illustration to Construction 4.11. Every solution contains all thick blue edges.

4.3 Intractability

4.3.1 On Series-Parallel Graphs with Vertex Cover Number Two

We show that $(2,2)$-Closed GBP is NP-hard even on series-parallel graphs with vertex cover number at most two. Additionally, we exclude the existence of polynomial kernels for $(2,2)$-Closed GBP regarding some parameterization. We start by proving the following proposition.

Proposition 4.10. $(2,2)$-Closed GBP *is* NP-*hard even on series-parallel graphs with vertex cover number two.*

We give a polynomial-time reduction from Hitting Set.

Construction 4.11. Let $\mathcal{I} = (U, \mathcal{F}, k)$ be an instance of Hitting Set with $U = \{1, \dots, n\}$ and $\mathcal{F} = \{F_1, \dots, F_m\}$. Construct an instance $\mathcal{I}' = (G, \mathcal{H}, k')$ of $(2,2)$-Closed GBP with $k' := n + k + 1$ as follows (see Figure 4.2 for an illustration).

Let G be initially empty. Add the vertex set $V_U := \{x_i \mid i \in U\}$ and the three vertices s, s', and t. Moreover, add the edge sets $E^* := \{\{s, s'\}\} \cup \bigcup_{i=1}^{n} \{\{s', x_i\}\}$ and $E_U := \bigcup_{i=1}^{n} \{\{x_i, t\}\}$. Finally, let $\mathcal{H} := E^* \cup \{Z_1, \dots, Z_m\}$ with $Z_j := \{s, t\} \cup \bigcup_{i \in F_j} \{x_i\}$ for every $F_j \in \mathcal{F}$. $\diamond$

We make two observations.

Observation 4.12. *The graph G constructed in Construction 4.11 is series-parallel and has vertex cover number two.*

Observation 4.13. *Let $\mathcal{I}'$ be a* `yes`*-instance obtained by Construction 4.11. Then, every solution F contains all edges in E^*.*

By the following lemma, we prove Proposition 4.10.

Lemma 4.14. *Let $\mathcal{I}'$ be the instance of* $(2,2)$-Closed GBP *obtained from an instance $\mathcal{I}$ of* Hitting Set *using Construction 4.11. Then, $\mathcal{I}$ is a* `yes`*-instance if and only if $\mathcal{I}'$ is a* `yes`*-instance.*

Proof. ($\Rightarrow$) Let $U' \subseteq U$ be a solution to $\mathcal{I}$. We claim that $F := E^* \cup \bigcup_{i \in U'} \{\{x_i, t\}\}$ is a solution to $\mathcal{I}'$. Note that $|F| \leq n+k+1$. Since $E^* \subseteq F$, it holds that $\operatorname{diam}(G[F]^2[H]) \leq 2$ for every habitat $H \in E^*$. Assume towards a contradiction that there is a habitat $Z_j \in \mathcal{H}$ such that $\operatorname{diam}(G[F]^2[Z_j]) > 2$. Because of $E^* \subseteq F$, it holds that $G[F]^2[Z_j \setminus \{t\}]$ is a clique. It follows that no edge from $\{\{x_i, t\} \mid i \in F_j\}$ is included in F. Therefore, $F_j \cap U' = \emptyset$, a contradiction to U' being a solution to $\mathcal{I}$.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. We claim that $U' := \{i \in U \mid \{x_i, t\} \in F\}$ is a solution to $\mathcal{I}$. Note that it follows from $E^* \subseteq F$ that $|U'| \leq k$. Assume towards a contradiction that U' is not a solution. Then, there is an $F_j \in \mathcal{F}$ with $F_j \cap U' = \emptyset$. Thus, no edge from $\{\{x_i, t\} \mid i \in F_j\}$ is included in F. It follows that the vertex t is isolated in $G[F][Z_j]$. Since for every $v \in Z_j \setminus \{t\}$ there is no (v, t)-path of length two in G, it holds that t is also isolated in $G[F]^2[Z_j]$. This implies that $\operatorname{diam}(G[F]^2[Z_j]) = \infty$, a contradiction to F being a solution to $\mathcal{I}'$. □

Finally, we discuss polynomial kernels. Hitting Set parameterized by the size of the universe U does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly [DLS14]. Since Construction 4.11 is a polynomial parameter transformation from Hitting Set parameterized by $|U|$ to $(2,2)$-Closed GBP parameterized by solution size k, we get the following.

Proposition 4.15. *Unless* NP $\subseteq$ coNP/poly*,* $(2,2)$-Closed GBP *parameterized by k admits no polynomial kernel.*

Since the graph G constructed in Construction 4.11 only has $|U|+3$ vertices, virtually every natural graph parameter of G is polynomially bounded by $|U|$. This implies the following.

Proposition 4.16. *Unless* NP $\subseteq$ coNP/poly*,* $(2,2)$-Closed GBP *parameterized by virtually any natural graph parameter admits no polynomial kernel.*

4.3.2 On Graphs with Distance to Clique Two

In this section, we show the following.

Proposition 4.17. $(2,2)$-Closed GBP *is* NP*-hard even on graphs with distance to clique two.*

We give a polynomial-time reduction from Hitting Set.

Construction 4.18. Let $\mathcal{I} = (U, \mathcal{F}, k)$ be an instance of Hitting Set with $U = \{1, \ldots, n\}$ and $\mathcal{F} = \{F_1, \ldots, F_m\}$. Construct an instance $\mathcal{I}' = (G, \mathcal{H}, k')$ of $(2,2)$-Closed GBP with $k' := n + k + 2$ as follows (see Figure 4.3 for an illustration).

Let G be initially empty. Add the vertex set $V_U := \{x_i \mid i \in U\}$ and the four vertices $s, s', t,$ and t'. Moreover, add the edge sets $E^* := \{\{s, s'\}, \{t', t\}\} \cup \bigcup_{i=1}^{n} \{\{s', x_i\}\}$, $E' := \{\{s', t'\}\} \cup [V_U]^2$, and $E_U := \bigcup_{i=1}^{n} \{\{x_i, t'\}\}$. Finally, let $\mathcal{H} := \mathcal{E} \cup \{Z_1, \ldots, Z_m\}$ where $\mathcal{E} := \{\{t', t\}\} \cup \bigcup_{i=1}^{n} \{\{s, x_i\}\}$ and $Z_j := \{s, t\} \cup \bigcup_{i \in F_j} \{x_i\}$ for every $F_j \in \mathcal{F}$. ⋄

We make two observations.

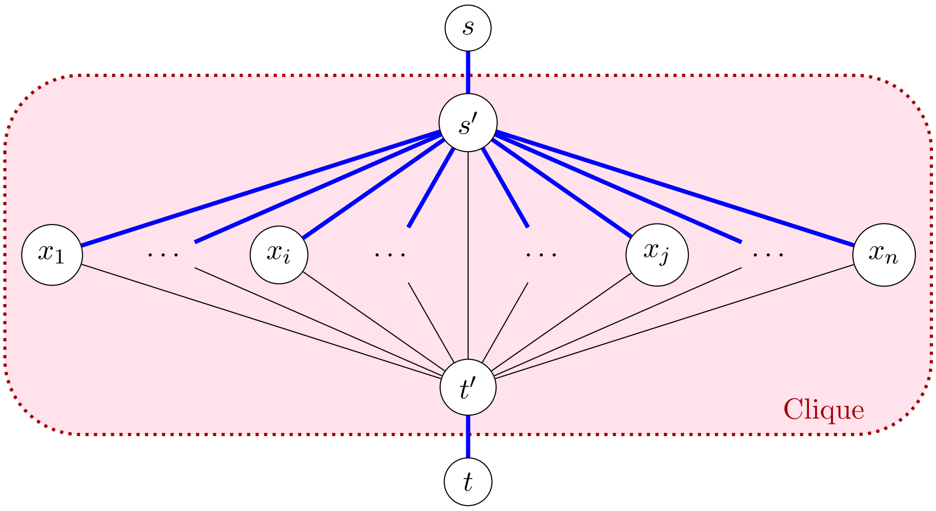


Figure 4.3: Illustration to Construction 4.18. Every solution contains all thick blue edges. Edges with both endvertices in V_U are not shown.

Observation 4.19. *The graph G constructed in Construction 4.18 has distance to clique two.*

Observation 4.20. *Let $\mathcal{I}'$ be a* `yes`*-instance obtained by Construction 4.18. Then, every solution F contains all edges in E^*.*

By the following lemma, we prove Proposition 4.17.

Lemma 4.21. *Let $\mathcal{I}'$ be the instance of* (2, 2)-Closed GBP *obtained from an instance $\mathcal{I}$ of* Hitting Set *using Construction 4.18. Then, $\mathcal{I}$ is a* `yes`*-instance if and only if $\mathcal{I}'$ is a* `yes`*-instance.*

Proof. ($\Rightarrow$) Let $U' \subseteq U$ be a solution to $\mathcal{I}$. We claim that $F := E^* \cup \bigcup_{i \in U'} \{x_i, t'\}$ is a solution to $\mathcal{I}'$. Note that $|F| \leq n+k+2$. Since $E^* \subseteq F$, it holds that $\operatorname{diam}(G[F]^2[H]) \leq 2$ for every habitat $H \in \mathcal{E}$. Assume towards a contradiction that there is a habitat $Z_j \in \mathcal{H}$ such that $\operatorname{diam}(G[F]^2[Z_j]) > 2$. Because of $E^* \subseteq F$, it holds that $G[F]^2[Z_j \setminus \{t\}]$ is a clique. It follows that no edge from $\{\{x_i, t\} \in E(G^2) \mid i \in F_j\}$ is included in $E(G[F]^2[Z_j])$. As $\{t', t\} \in F$, this implies that no edge from $\{\{x_i, t'\} \in E(G) \mid i \in F_j\}$ is included in F. Therefore, $F_j \cap U' = \emptyset$, a contradiction to U' being a solution to $\mathcal{I}$.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. We claim that $U' := \{i \in U \mid \{x_i, t'\} \in F\}$ is a solution to $\mathcal{I}$. Note that it follows from $E^* \subseteq F$ that $|U'| \leq k$. Assume towards a contradiction that U' is not a solution. Then, there is an $F_j \in \mathcal{F}$ with $F_j \cap U' = \emptyset$. Hence, no edge from the edge set $E_j := \{\{x_i, t'\} \mid i \in F_j\}$ is included in F. Let $v \in Z_j \setminus \{t\}$ be a vertex. There is at most one (v, t)-path of length at most two in G. As this path (if existent) contains an edge from E_j, it follows that $\{v, t\} \notin E(G[F]^2[Z_j])$. Hence, t is isolated in $G[F]^2[Z_j]$. This implies that $\operatorname{diam}(G[F]^2[Z_j]) = \infty$, a contradiction to F being a solution to $\mathcal{I}'$. $\square$

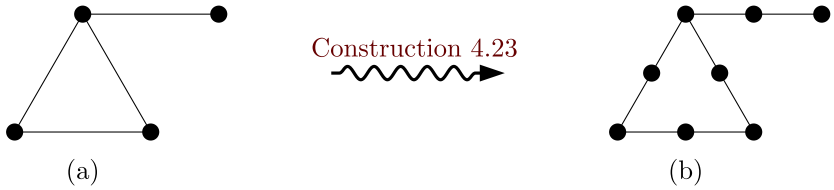


Figure 4.4: Illustration to Construction 4.23. (a) Example graph G of the input instance $\mathcal{I} = (G, \mathcal{H}, k)$. (b) The graph G' obtained from G using Construction 4.23.

4.3.3 On Bipartite Graphs with Only One Habitat

In this section, we show the following.

Proposition 4.22. *$(2,2)$-*Closed GBP *is* NP*-hard even on bipartite graphs where $\mathcal{H}$ contains exactly one habitat.*

Fluschnik and Kellerhals [FK21] show that 2-Diam GBP is NP-hard even if $|\mathcal{H}| = 1$. We give a polynomial-time reduction from 2-Diam GBP on instances with $|\mathcal{H}| = 1$ to $(2,2)$-Closed GBP.

Construction 4.23. Let $\mathcal{I} = (G, \mathcal{H}, k)$ be an instance of 2-Diam GBP with $|\mathcal{H}| = 1$. Construct an instance $\mathcal{I}' = (G', \mathcal{H}', k')$ of $(2,2)$-Closed GBP with $k' := 2k$ as follows (see Figure 4.4 for an illustration).

Create the graph G' by subdividing each edge in G. More precisely, let G' be initially equal to G and do the following for each edge $e = \{x, y\} \in E(G)$. Delete e from $E(G')$, add a new vertex v_e to $V(G')$, and add the edges $\{x, v_e\}$ and $\{v_e, y\}$ to $E(G')$. Finally, let $\mathcal{H}' := \mathcal{H}$.

Observation 4.24. *The graph G' constructed in Construction 4.23 is bipartite.*

Lemma 4.25. *Let $\mathcal{I}'$ be the instance of* $(2,2)$-Closed GBP *obtained from an instance $\mathcal{I}$ of* 2-Diam GBP *using Construction 4.23. Then, $\mathcal{I}$ is a* `yes`*-instance if and only if $\mathcal{I}'$ is a* `yes`*-instance.*

Proof. ($\Rightarrow$) Let F be a solution to $\mathcal{I}$. We claim that $F' := \{\{x, v_e\}, \{v_e, y\} \mid e = \{x, y\} \in F\}$ is a solution to $\mathcal{I}'$. Note that $|F'| \leq 2k$. Let $H \in \mathcal{H}$ and let $x, y \in H$. It suffices to show that $\operatorname{dist}_{G'[F']^2[H]}(x, y) \leq 2$. Since F is a solution to $\mathcal{I}$, it holds that $\operatorname{dist}_{G[F][H]}(x, y) \leq 2$. Thus, there is an (x, y)-path P of length at most two in $G[F][H]$. We consider an edge $e = \{a, b\} \in E(P)$. As $e \in F$, it follows that $\{a, v_e\}, \{v_e, b\} \in F'$. This implies that $e \in E(G'[F']^2)$. Therefore, P is also an (x, y)-path in $G'[F']^2[H]$. Hence, $\operatorname{dist}_{G'[F']^2[H]}(x, y) \leq 2$.

($\Leftarrow$) Let F' be a solution to $\mathcal{I}'$. We claim that $F := \{e = \{x, y\} \mid \{x, v_e\}, \{v_e, y\} \in F'\}$ is a solution to $\mathcal{I}$. Note that $|F| \leq k$. Again, let $H \in \mathcal{H}$ and let $x, y \in H$. It suffices to show that $\operatorname{dist}_{G[F][H]}(x, y) \leq 2$. Since F' is a solution to $\mathcal{I}'$, it holds that $\operatorname{dist}_{G'[F']^2[H]}(x, y) \leq 2$. Thus, there is an (x, y)-path P of length at most two in $G'[F']^2[H]$. We consider an edge $e = \{a, b\} \in E(P)$. As $e \in E(G'[F']^2)$, it follows that $\{a, v_e\}, \{v_e, b\} \in F'$. This implies that $\{a, b\} \in F$. Therefore, P is also an (x, y)-path in $G[F][H]$. Hence, $\operatorname{dist}_{G[F][H]}(x, y) \leq 2$. $\square$

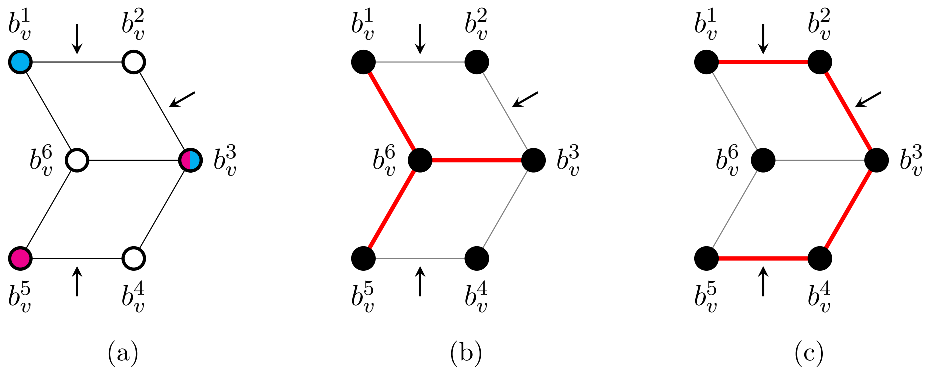


Figure 4.5: Illustration of vertex gadget B_v with habitat set $\mathcal{H}_v$ defined in Definition 4.27. Docking edges have arrows pointing to them. (a) The vertices of each of the two habitats are marked by two different colors. Vertices that are not included in any habitat are left white. (b) The thick red edges denote the unique solution of minimum size to $(B_v, \mathcal{H}_v)$. (c) The thick red edges denote a solution to $(B_v, \mathcal{H}_v)$ that is not minimum but includes all docking edges.

4.3.4 On Planar Graphs with Habitats of Size at Most Two

In this section, we show the following.

Proposition 4.26. (2, 2)-Closed GBP *is* NP*-hard even on planar graphs of maximum degree four with each habitat having size at most two.*

We give a polynomial-time reduction from Planar Cubic Vertex Cover.

For easier notation, we denote an instance of the optimization version of (2, 2)-Closed GBP with graph G and habitat set $\mathcal{H}$ by $(G, \mathcal{H})$ without explicitly stating that we interpret $(G, \mathcal{H})$ as an instance of the optimization version of (2, 2)-Closed GBP and not, say, any other problem.

In the upcoming reduction, we replace each vertex of the given instance of Planar Cubic Vertex Cover by a graph ("vertex gadget") and each edge by a graph ("edge gadget"). These gadgets mimic, in a sense, the nodes and edges of the graph of the Planar Cubic Vertex Cover-instance. We define the gadgets in preparation for the reduction.

Definition 4.27. Let G be a graph and let $v \in V(G)$ be a vertex. The vertex gadget B_v corresponding to v is the graph with vertex set $V(B_v) := \{b_v^1, \ldots, b_v^6\}$ and edge set $E(B_v) := \{\{b_v^1, b_v^6\}, \{b_v^3, b_v^6\}\} \cup \bigcup_{i=1}^{5} \{\{b_v^i, b_v^{i+1}\}\}$. We define the habitat set for B_v to be $\mathcal{H}_v := \{\{b_v^1, b_v^3\}, \{b_v^3, b_v^5\}\}$. The *docking edges* of B_v are $\{b_v^1, b_v^2\}$, $\{b_v^2, b_v^3\}$, and $\{b_v^4, b_v^5\}$. If a vertex is an endvertex of a docking edge, then we call it a *docking vertex*. (See Figure 4.5a for an illustration.)

Docking edges are used to "glue together" vertex and edge gadgets in the reduction. (The choice of the docking edges is somewhat arbitrary. Because of symmetry, we could use $\{b_v^3, b_v^4\}$ instead of $\{b_v^2, b_v^3\}$ as a docking edge.)

Each instance $(B_v, \mathcal{H}_v)$ has the unique minimum solution shown in Figure 4.5b. The minimum solution does not contain any of the docking edges. However, as shown in

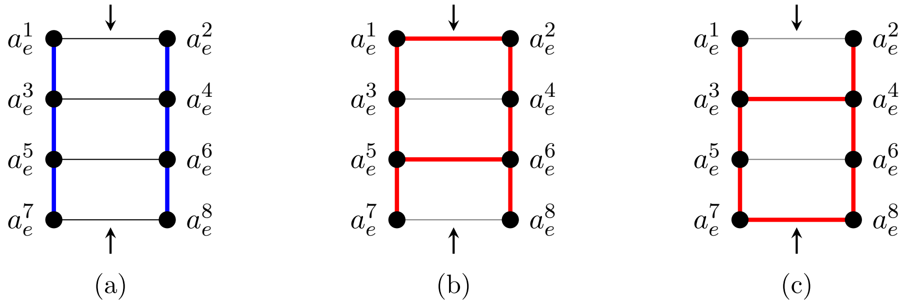


Figure 4.6: Illustration of edge gadget A_e with habitat set $\mathcal{H}_e$ defined in Definition 4.28. Docking edges have arrows pointing to them. (a) Every solution contains all thick blue edges. (b)&(c) In each subfigure the thick red edges denote a solution of minimum size to $(A_e, \mathcal{H}_e)$.

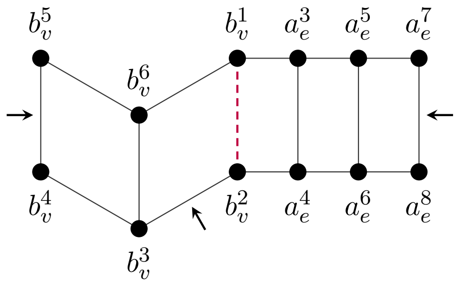


Figure 4.7: Result of gluing together a vertex gadget B_v and an edge gadget A_e using the docking edges $\{b_v^1, b_v^2\}$ and $\{a_e^1, a_e^2\}$. The used docking edge is dashed and purple. Unused docking edges have arrows pointing to them.

Figure 4.5c, there is a solution that contains all docking edges and is only by one edge larger than the minimum solution.

Definition 4.28. Let G be a graph and let $e \in E(G)$ be an edge. The edge gadget A_e corresponding to e is the graph with vertex set $V(A_e) := \{a_e^1, \dots, a_e^8\}$ and edge set $E(A_e) := E^* \cup \{\{a_e^1, a_e^2\}, \{a_e^3, a_e^4\}, \{a_e^5, a_e^6\}, \{a_e^7, a_e^8\}\}$ where $E^* := \bigcup_{i=1}^{6}\{\{a_e^i, a_e^{i+2}\}\}$. The habitat set for A_e is $\mathcal{H}_e := E^* \cup \{\{a_e^1, a_e^4\}, \{a_e^3, a_e^6\}, \{a_e^5, a_e^8\}\}$. The docking edges of A_e are $\{a_e^1, a_e^2\}$ and $\{a_e^7, a_e^8\}$. (See Figure 4.6a for an illustration.)

Each instance $(A_e, \mathcal{H}_e)$ has multiple minimum solutions. For each docking edge there is a minimum solution containing it as shown in Figures 4.6b and 4.6c. There is no minimum solution containing both docking edges.

Gluing together a vertex gadget B_v and an edge gadget A_e means to select a previously unused docking edge $\{b_v^i, b_v^j\} \in E(B_v)$ and a previously unused docking edge $\{a_e^k, a_e^\ell\} \in E(A_e)$ and to set $a_e^k := b_v^i$ and $a_e^\ell := b_v^j$ (see Figure 4.7 for an illustration).

Construction 4.29. Let $\mathcal{I} = (G, k)$ be an instance of Planar Cubic Vertex Cover with $V(G) = \{v_1, \dots, v_n\}$, and $E(G) = \{e_1, \dots, e_m\}$. Construct an instance $\mathcal{I}' = (G', \mathcal{H}', k)$ of (2, 2)-Closed GBP with $k' := 3n + 7m + k$ as follows (see Figure 4.8 for an illustration).

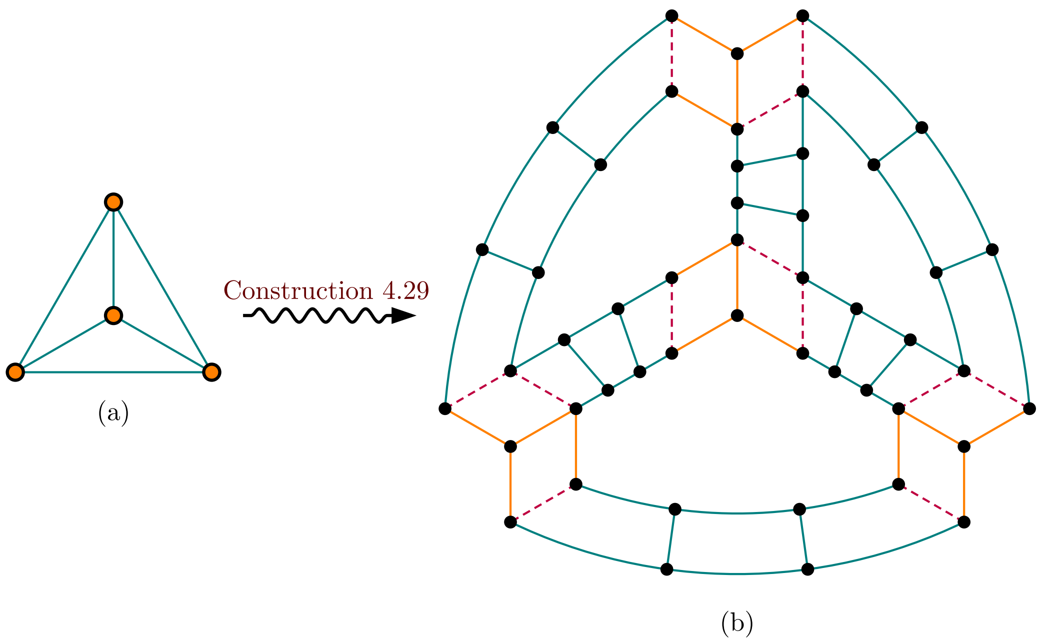


Figure 4.8: Illustration to Construction 4.29. (a) Example graph G of the input instance $\mathcal{I} = (G, k)$. Vertices are orange and edges are turquoise. (b) A graph G' that can be obtained from G using Construction 4.29. Docking edges (being both part of a vertex and an edge gadget) are dashed and purple. Edges that are only part of a vertex gadget are orange. Edges that are only part of an edge gadget are turquoise.

Let G' and $\mathcal{H}'$ be initially empty. Whenever we make an addition to G', we assume that we make the addition in a reasonable manner preserving the planarity of G'. For each $v \in V(G)$ add the vertex gadget B_v to G' and extend $\mathcal{H}$ by $\mathcal{H}_v$. Then, follow the steps below for each $\{u, v\} \in E(G)$. Select a docking edge e_u of B_u and a docking edge e_v of B_v such that both docking edges have not been used before. Add A_e to G' such that A_e is glued to both B_u and B_v using the selected docking edges e_u and e_v. Moreover, extend $\mathcal{H}$ by $\mathcal{H}_e$. $\diamond$

Observation 4.30. *The graph G' constructed in Construction 4.29 is planar and has maximum degree at most four.*

Observation 4.31. *The set $\mathcal{H}'$ constructed in Construction 4.29 only contains habitats of size two.*

The following observation says that the habitats of the constructed instance $\mathcal{I}'$ need to be connected "locally", i.e., for each habitat $H \subseteq \mathcal{H}'$ only edges of the gadget containing H are useful for connecting H. This observation is important as it implies that in the constructed graph G' there are no "shortcuts" to the previously discussed small solutions for single vertex and edge gadgets.

Observation 4.32. *Let $Y \subseteq G'$ be a gadget (vertex gadget or edge gadget) and let $\{u, v\} \in \mathcal{H}'$ be a habitat with $u, v \in V(Y)$. Then, it holds for every (u, v)-path of length at most two in G' that $P \subseteq Y$.*

Given a vertex gadget $B_v \subseteq G'$, we denote the set of docking edges of B_v by $E_{\text{dock}}(B_v)$. Likewise, given an edge gadget $A_e \subseteq G'$, we denote the set of docking edges of A_e by $E_{\text{dock}}(A_e)$. Furthermore, we use the notation $E_{\text{in}}(A_e) := E(A_e) \setminus E_{\text{dock}}(A_e)$ to refer to the set of non-docking edges of A_e.

The following three observations are direct consequences of facts mentioned earlier.

Observation 4.33. *Let $A_e \subseteq G'$ be an edge gadget and let $e' \in E_{\text{dock}}(A_e)$ be a docking edge of A_e. There is a subset $F_e \subseteq E_{\text{in}}(A_e)$ with $|F_e| \leq 7$ such that $F_e \cup \{e'\}$ is a solution to $(A_e, \mathcal{H}_e)$.*

Observation 4.34. *Let F be a solution to the constructed instance $\mathcal{I}'$ and let $B_v \subseteq G'$ be a vertex gadget. Then,*

$$|F \cap E(B_v)| \geq \begin{cases} 3, & \textit{if } F \cap E_{\text{dock}}(B_v) = \emptyset, \\ 4, & \textit{otherwise.} \end{cases}$$

Observation 4.35. *Let F be a solution to the constructed instance $\mathcal{I}'$ and let $A_e \subseteq G'$ be an edge gadget. Then,*

$$|F \cap E_{\text{in}}(A_e)| \geq \begin{cases} 8, & \textit{if } F \cap E_{\text{dock}}(A_e) = \emptyset, \\ 7, & \textit{otherwise.} \end{cases}$$

Lemma 4.36. *Let $\mathcal{I}'$ be the instance of* $(2,2)$-Closed GBP *obtained from an instance $\mathcal{I}$ of* Planar Cubic Vertex Cover. *Then, $\mathcal{I}$ is a* `yes`*-instance if and only if $\mathcal{I}'$ is a* `yes`*-instance.*

Proof. ($\Rightarrow$) Let $S \subseteq V(G)$ be a vertex cover of G with size at most k. Let $\bar{S} := V(G) \setminus S$. The set

$$F_{\bar{S}} := \bigcup_{v \in \bar{S}} \left\{ \{b_v^1, b_v^6\}, \{b_v^3, b_v^6\}, \{b_v^5, b_v^6\} \right\}$$

is a union of local solutions as depicted in Figure 4.5b. The set

$$F_S := \bigcup_{v \in S} \left(\bigcup_{i=1}^{4} \left\{ \{b_v^i, b_v^{i+1}\} \right\} \right)$$

is a union of local solutions as depicted in Figure 4.5c. Let $e \in E(G)$ be an edge. Since S is a vertex cover, at least one of the endvertices of e is contained in S. This implies that at least one of the docking edges of A_e is included in F_S. Thus, by Observation 4.33 there is a set F_e of size at most seven such that $(F_e \cup F_S) \cap E(A_e)$ is a solution to $(A_e, \mathcal{H}_e)$. We claim that $F := F_{\bar{S}} \cup F_S \cup \bigcup_{i=1}^{m} F_{e_j}$ is a solution to $\mathcal{I}'$. It holds that $|F| \leq 3 \cdot (n-k) + 4k + 7m = k'$. Since F is a union of local solutions, it holds that $\operatorname{diam}(G[F][H]) \leq 2$ for every $H \in \mathcal{H}$.

($\Leftarrow$) Let F be a solution to $\mathcal{I}'$. Let $S' := \{v \in V(G) \mid F \cap E_{\text{dock}}(B_v) \neq \emptyset\ \}$ and let S'' be a set constructed as follows. For each edge $e \in E(G)$ with $e \cap S' = \emptyset$ add one of the two endvertices of e to S''. Clearly, $S := S' \cup S''$ is a vertex cover of G. It is left to show that $|S| \leq k$. Let $E_V := \bigcup_{i=1}^{n} E(B_{v_i})$. By Observation 4.34 it holds that $|F \cap E_V| \geq 4 \cdot |S'| + 3 \cdot (n - |S'|) = 3n + |S'|$. Let $E_E := \bigcup_{i=1}^{m} E_{\text{in}}(A_{e_i})$. Let ℓ be the number of edge gadgets $A_e \subseteq G'$ with the property that $F \cap E_{\mathrm{dock}}(A_e) = \emptyset$. By Observation 4.35 it holds that $|F \cap E_E| \geq 8\ell + 7 \cdot (m - \ell) = 7m + \ell$. If for an edge $e \in E(G)$ it holds that $e \cap S' = \emptyset$, then $F \cap E_{\mathrm{dock}}(A_e) = \emptyset$. By construction of S'', this implies that $\ell \geq |S''|$. It follows that $|F \cap E_E| \geq 7m + |S''|$. Hence, $|F| = |F \cap E_V| + |F \cap E_E| \geq 3n + |S'| + 7m + |S''|$. By rearrangement we get $|S| = |S'| + |S''| \leq |F| - 3n - 7m \leq k$. $\square$

Chapter 5

Epilogue

We studied the problems 2-Diam GBP and $(2,2)$-Closed GBP. A strong hardness result we obtained, especially when considering the most probable structure of real-world data, is that 2-Diam GBP is NP-hard even on planar graphs of maximum degree five with each habitat having size at most four. The situation is even worse for $(2,2)$-Closed GBP, as we found that $(2,2)$-Closed GBP is NP-hard even on planar graphs of maximum degree four with each habitat having size at most two.

Nevertheless, we also provided polynomial-time algorithms for some cases, most notably the cases of 2-Diam GBP-C where each habitat has size at most three and the input graph is planar or has maximum degree at most four. Moreover, we showed that both 2-Diam GBP-C and $(2,2)$-Closed GBP-C are fixed-parameter tractable with respect to the feedback edge number.

For the problems discussed in this thesis, we note that their computational hardness in quite heavily restricted cases can mean that they become equivalent under constraints that allow for polynomial-time solvability.

As for future work, we most prominently leave open the question of whether 2-Diam GBP-C can be solved in polynomial time on graphs of maximum degree at most four. We conjecture that this case is indeed solvable in polynomial time. The idea is to define areas similar to how we did for the case of maximum degree at most three in Section 3.2.1. Unlike Section 3.2.1, we do not generate the global solution by naively combining local solutions. Instead, we assume that the connections between areas have a tree-like structure, allowing us to construct a global solution by dynamic programming. The details of this, however, seem challenging.

Moreover, it seems worthwhile to consider the parameterized complexity of 2-Diam GBP-C with respect to the combined parameter treewidth plus maximum degree $\omega+\Delta$. We claim that this parameterization results in fixed-parameter tractability. Given a tree decomposition $(T, \mathcal{X})$ of the input graph G, the idea is to replace each bag $X_t \in \mathcal{X}$ by its 2-neighborhood in G. The result also is a tree decomposition of G and the factor by which its width increases is upper bounded by a function only depending on Δ. Having done this, for each habitat there is a bag containing it, or otherwise the input instance is a `no`-instance. This enables us to construct a solution using dynamic programming. We further claim that by being more generous and replacing each bag by its 4-neighborhood instead of its 2-neighborhood, the approach can also be used to show fixed-parameter tractability with respect to $\omega + \Delta$ for $(2,2)$-Closed GBP-C.

With a formal proof of the above claim added to our results, the question of whether 2-Diam GBP-C and $(2,2)$-Closed GBP-C admit fixed-parameter tractable algorithms with respect to most of the graph parameters commonly studied would be settled (assuming $\mathrm{P} \neq \mathrm{NP}$). Nevertheless, many interesting parameterizations remain to be explored. Especially in view of the likely structure of actual inputs, 2-Diam GBP on planar graphs parameterized by the number of habitats seems to be one of them.

Bibliography

[AAR10] Dana Angluin, James Aspnes, and Lev Reyzin. *Inferring Social Networks from Outbreaks*. In: *Proceedings of the Twenty-First International Conference on Algorithmic Learning Theory, ALT 2010*. Springer, 2010, pp. 104–118 (cit. on p. 11).

[Aga+13] Deepesh Agarwal, Julio-Cesar Silva Araujo, Christelle Caillouet, Frederic Cazals, David Coudert, and Stephane Pérennes. *Connectivity Inference in Mass Spectrometry Based Structure Determination*. In: *Proceedings of the Twenty-First European Symposium on Algorithms, ESA 2013*. Springer, 2013, pp. 289–300 (cit. on p. 11).

[Bar+13] Hugo Barbalho, Isabel Rosseti, Simone L. Martins, and Alexandre Plastino. *A Hybrid Data Mining GRASP with Path-Relinking*. In: *Computers & Operations Research* 40.12 (2013), pp. 3159–3173 (cit. on p. 13).

[Ben03] Andrew Bennett. *Linkages in the Landscape; The Role of Corridors and Connectivity in Wildlife Conservation*. Second edition. International Union for Conservation of Nature and Natural Resources, 2003 (cit. on p. 9).

[BHM11] Mohammadhossein Bateni, Mohammadtaghi Hajiaghayi, and Dániel Marx. *Approximation Schemes for Steiner Forest on Planar Graphs and Graphs of Bounded Treewidth*. In: *Journal of the ACM* 58.5 (2011), pp. 1–37 (cit. on p. 11).

[Bra+12] Ulrik Brandes, Sabine Cornelsen, Barbara Pampel, and Arnaud Sallaberry. *Path-Based Supports for Hypergraphs*. In: *Journal of Discrete Algorithms* 14 (2012), pp. 248–261 (cit. on p. 11).

[Bro+15] Jedediah F. Brodie, Anthony J. Giordano, Brett Dickson, Mark Hebblewhite, Henry Bernard, Jayasilan Mohd-Azlan, Jesse Anderson, and Laurentius Ambu. *Evaluating Multispecies Landscape Connectivity in a Threatened Tropical Mammal Community*. In: *Conservation Biology* 29.1 (2015), pp. 122–132 (cit. on p. 9).

[BTY11] Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. *Kernel Bounds for Disjoint Cycles and Disjoint Paths*. In: *Theoretical Computer Science* 412.35 (2011), pp. 4570–4578 (cit. on p. 21).

[CDS04] Vincent Conitzer, Jonathan Derryberry, and Tuomas Sandholm. *Combinatorial Auctions with Structured Item Graphs*. In: *Proceedings of the Nineteenth AAAI Conference on Artificial Intelligence, AAAI 2004*. AAAI Press, 2004, pp. 212–218 (cit. on p. 11).

[Che+15] Jiehua Chen, Christian Komusiewicz, Rolf Niedermeier, Manuel Sorge, Ondřej Suchý, and Mathias Weller. *Polynomial-Time Data Reduction for the Subset Interconnection Design Problem*. In: *SIAM Journal on Discrete Mathematics* 29.1 (2015), pp. 1–25 (cit. on p. 11).

[Cho+07] Gregory Chockler, Roie Melamed, Yoav Tock, and Roman Vitenberg. *Constructing Scalable Overlays for Pub-Sub with Many Topics*. In: *Proceedings of the Twenty-Sixth ACM Symposium on Principles of Distributed Computing, PODC 2007*. ACM, 2007, pp. 109–118 (cit. on p. 11).

[CK15] Mark R. Christie and L. Lacey Knowles. *Habitat Corridors Facilitate Genetic Resilience Irrespective of Species Dispersal Abilities or Population Sizes*. In: *Evolutionary Applications* 8.5 (2015), pp. 454–463 (cit. on p. 9).

[Clí+19] Glaubos Clímaco, Isabel Rosseti, Luidi Simonetti, and Marcos Guerine. *Combining Integer Linear Programming with a State-Of-The-Art Heuristic for the 2-Path Network Design Problem*. In: *International Transactions in Operational Research* 26.2 (2019), pp. 615–641 (cit. on p. 13).

[Cyg+15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015 (cit. on p. 17).

[DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013 (cit. on p. 17).

[Die17] Reinhard Diestel. *Graph Theory*. Fifth edition. Vol. 173. Graduate Texts in Mathematics. Springer, 2017 (cit. on p. 17).

[DJ04] Geir Dahl and Bjarne Johannessen. *The 2-Path Network Problem*. In: *Networks* 43.3 (2004), pp. 190–199 (cit. on pp. 13, 16).

[DK95] Ding-Zhu Du and Dean F. Kelley. *On Complexity of Subset Interconnection Designs*. In: *Journal of Global Optimization* 6 (1995), pp. 193–205 (cit. on p. 11).

[DLS14] Michael Dom, Daniel Lokshtanov, and Saket Saurabh. *Kernelization Lower Bounds Through Colors and IDs*. In: *ACM Transactions on Algorithms* 11.2 (2014), 13:1–13:20 (cit. on pp. 58, 63).

[DPL21] Svenja Dobbert, Roland Pape, and Jörg Löffler. *Continent-Wide Modelling of Potential Habitats and Connectivity for the Eurasian Lynx (Lynx Lynx) in Europe*. In: *Erdkunde* 75.2 (2021), pp. 69–86 (cit. on p. 9).

[Fan+08] Hongbing Fan, Christian Hundt, Yu-Liang Wu, and Jason Ernst. *Algorithms and Implementation for Interconnection Graph Problem*. In: *Proceedings of the Second International Conference on Combinatorial Optimization and Applications, COCOA 2008*. Springer, 2008, pp. 201–210 (cit. on pp. 11, 12).

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006 (cit. on p. 17).

[FK21] Till Fluschnik and Leon Kellerhals. *Placing Green Bridges Optimally, with a Multivariate Analysis*. In: *Connecting with Computability. Proceedings of the Seventeenth Conference on Computability in Europe, CiE 2021*. Springer, 2021, pp. 204–216 (cit. on pp. 9–13, 15, 56, 65).

[Fra+15] Fabrizio Frati, Serge Gaspers, Joachim Gudmundsson, and Luke Mathieson. *Augmenting Graphs to Minimize the Diameter*. In: *Algorithmica* 72 (2015), pp. 995–1010 (cit. on p. 11).

[Gio+17] Aristides Gionis, Polina Rozenshtein, Nikolaj Tatti, and Evimaria Terzi. *Community-Aware Network Sparsification*. In: *Proceedings of the Seventeenth SIAM International Conference on Data Mining, SDM 2017*. SIAM, 2017, pp. 426–434 (cit. on p. 12).

[Gou98] Luis Gouveia. *Using Variable Redefinition for Computing Lower Bounds for Minimum Spanning and Steiner Trees with Hop Constraints*. In: *INFORMS Journal on Computing* 10.2 (1998), pp. 180–188 (cit. on p. 11).

[Her21] Maike Herkenrath. *The Influence of Habitat Structure on the Algorithmic Complexity of Placing Green Bridges*. Bachelor Thesis. TU Berlin, 2021 (cit. on p. 11).

[Her+22] Maike Herkenrath, Till Fluschnik, Francesco Grothe, and Leon Kellerhals. *Placing Green Bridges Optimally, with Habitats Inducing Cycles*. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 22*. IJCAI Organization, 2022, pp. 3825–3831 (cit. on pp. 11–13, 15, 41, 42, 48, 55, 56).

[Her22] Emanuel Herrendorf. *On the Complexity of Community-Aware Network Sparsification*. Master Thesis. University of Marburg, 2022 (cit. on p. 12).

[Hos+12] Jun Hosoda, Juraj Hromkovič, Taisuke Izumi, Hirotaka Ono, Monika Steinová, and Koichi Wada. *On the Approximability and Hardness of Minimum Topic Connected Overlay and Its Special Instances*. In: *Theoretical Computer Science* 429 (2012), pp. 144–154 (cit. on p. 12).

[HZ09] Nicole E. Heller and Erika S. Zavaleta. *Biodiversity Management in the Face of Climate Change: A Review of 22 Years of Recommendations*. In: *Biological Conservation* 142.1 (2009), pp. 14–32 (cit. on p. 9).

[JLL21] Jesper Jansson, Christos Levcopoulos, and Andrzej Lingas. *Online and Approximate Network Construction from Bounded Connectivity Constraints*. In: *Proceedings of the Twelfth International Conference on Algorithms and Complexity, CIAC 2021*. Springer, 2021, pp. 314–325 (cit. on p. 12).

[KS03] Ephraim Korach and Michal Stern. *The Clustering Matroid and the Optimal Clustering Tree*. In: *Mathematical Programming* 98.1 (2003), pp. 385–414 (cit. on p. 12).

[KS08] Ephraim Korach and Michal Stern. *The Complete Optimal Stars-Clustering-Tree Problem*. In: *Discrete Applied Mathematics* 156.4 (2008), pp. 444–450 (cit. on p. 12).

[KW21] Linda Kopaniak and Hannah Weber. *Luchs Lias im Oberen Donautal – Erste Erkenntnisse der Besenderung*. Forstliche Versuchs- und Forschungsanstalt Baden-Württemberg. 2021. URL: https://www.fva-bw.de/fileadmin/user_upload/Abteilungen/Wald_und_Gesellschaft/Wildtieroekologie/Grosse_Beutegreifer/210330_Artikel_Lias_FVA.pdf (visited on Mar. 25, 2023) (cit. on p. 9).

[Lai+11] Katherine J. Lai, Carla P. Gomes, Michael K. Schwartz, Kevin S. McKelvey, David E. Calkin, and Claire A. Montgomery. *The Steiner Multigraph Problem: Wildlife Corridor Design for Multiple Species*. In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011*. AAAI Press, 2011, pp. 1357–1364 (cit. on p. 11).

[LeB+13] Ronan LeBras, Bistra Dilkina, Yexiang Xue, Carla P. Gomes, Kevin S. McKelvey, Michael K. Schwartz, and Claire A. Montgomery. *Robust Network Design for Multispecies Conservation*. In: *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2013*. AAAI Press, 2013, pp. 1305–1312 (cit. on p. 11).

[Mei+20] Holger Meinig, Peter Boye, Michael Dähne, Rainer Hutterer, and Johannes Lang. *Rote Liste und Gesamtartenliste der Säugetiere (Mammalia) Deutschlands*. Stand November 2019. BfN-Schriftenvertrieb im Landwirtschaftsverlag, 2020 (cit. on p. 9).

[Moh01] Bojan Mohar. *Face Covers and the Genus Problem for Apex Graphs*. In: *Journal of Combinatorial Theory, Series B* 82.1 (2001), pp. 102–117 (cit. on p. 48).

[MVM10] Matthew F. Mccollister and Frank T. Van Manen. *Effectiveness of Wildlife Underpasses and Fencing to Reduce Wildlife-Vehicle Collisions*. In: *The Journal of Wildlife Management* 74.8 (2010), pp. 1722–1731 (cit. on p. 9).

[New93] William D. Newmark. *The Role and Design of Wildlife Corridors with Examples from Tanzania*. In: *Ambio* 22.8 (1993), pp. 500–504 (cit. on p. 9).

[Nie06] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Vol. 31. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006 (cit. on p. 17).

[NS21] Paolo Nobili and Antonio Sassano. *An $\mathcal{O}(n^2 \log n)$ Algorithm for the Weighted Stable Set Problem in Claw-Free Graphs*. In: *Mathematical Programming* 186 (2021), pp. 409–437 (cit. on p. 33).

[ÖLD17] Kasim Öztoprak, Sina Layazali, and Erdogan Dogdu. *Topic Distribution Constant Diameter Overlay Design Algorithm (TD-CD-ODA)*. In: *Proceedings of the Eleventh IEEE International Conference on Semantic Computing, ICSC 2017*. IEEE, 2017, pp. 482–487 (cit. on p. 12).

[OR11] Melih Onus and Andréa W. Richa. *Minimum Maximum-Degree Publish-Subscribe Overlay Network Design*. In: *IEEE/ACM Transactions on Networking* 19.5 (2011), pp. 1331–1343 (cit. on p. 12).

[OR16] Melih Onus and Andréa W. Richa. *Parameterized Maximum and Average Degree Approximation in Topic-Based Publish-Subscribe Overlay Network Design*. In: *Computer Networks* 94 (2016), pp. 307–317 (cit. on p. 12).

[Ple81] J. Plesnik. *The Complexity of Designing a Network with Minimum Diameter*. In: *Networks* 11.1 (1981), pp. 77–85 (cit. on p. 11).

[RR02] Celso C. Ribeiro and Isabel Rosseti. *A Parallel GRASP Heuristic for the 2-Path Network Design Problem*. In: *Proceedings of the Eighth European Conference on Parallel Processing, Euro-Par 2002*. Springer, 2002, pp. 922–926 (cit. on p. 13).

[SBVL87] A. A. Schoone, H. L. Bodlaender, and J. Van Leeuwen. *Diameter Increase Caused by Edge Deletion*. In: *Journal of Graph Theory* 11.3 (1987), pp. 409–427 (cit. on p. 11).

[SRR15] Daniel J. Smith, Rodney van der Ree, and Carme Rosell. *Wildlife Crossing Structures*. In: *Handbook of Road Ecology*. Wiley, 2015. Chap. 21, pp. 172–183 (cit. on p. 9).