Placing Green Bridges Optimally, with Close-Range Habitats in Sparse Graphs

Christian Wallisch
Technische Universität Berlin,
Algorithmics and Computational
Complexity Group
Berlin, Germany
wallisch@campus.tu-berlin.de

Till Fluschnik
Humboldt-Universität zu Berlin,
Department of Computer Science,
Algorithm Engineering Group
Berlin, Germany
till.fluschnik@hu-berlin.de

Leon Kellerhals
Technische Universität Clausthal,
Computational Intelligence
Clausthal-Zellerfeld, Germany
leon.kellerhals@tu-clausthal.de

ABSTRACT

We study a network design problem motivated by the challenge of placing wildlife crossings to reconnect fragmented habitats of animal species, which is among the 17 goals towards sustainable development by the UN: Given a graph, whose vertices represent the fragmented habitat areas and whose edges represent possible green bridge locations (with costs), and the habitable vertex set for each species' habitat, the goal is to find the cheapest set of edges such that each species' habitat is sufficiently connected. We focus on the established variant where a habitat is considered sufficiently connected if it has diameter two in the solution and study its complexity in cases justified by our setting namely small habitat sizes on planar graphs and graphs of small maximum degree Δ. We provide efficient algorithms and NP-hardness results for different values of Δ and maximum habitat sizes on general and planar graphs.

KEYWORDS

wildlife crossings; network sparsification; graph diameter; sparse graphs; polynomial-time algorithms; NP-hardness

ACM Reference Format:
Christian Wallisch, Till Fluschnik, and Leon Kellerhals. 2026. Placing Green Bridges Optimally, with Close-Range Habitats in Sparse Graphs. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/OYZM4458

1 INTRODUCTION

Habitat fragmentation due to human-made infrastructure like roads or train tracks leads to wildlife-vehicle collisions [39], a threat to animals so severe that it has a significant impact on biodiversity [5]. Installing wildlife crossings like bridges, tunnels, ropes, etc., to connect habitat patches of different animal species is a cost-efficient and effective measure against wildlife-vehicle collisions [26, 27]. 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. It seems desirable to place green bridges such that every animal can traverse their habitat using only a few green bridges. A natural question is where to build green bridges to achieve dense connectivity of habitats while keeping financial costs low.

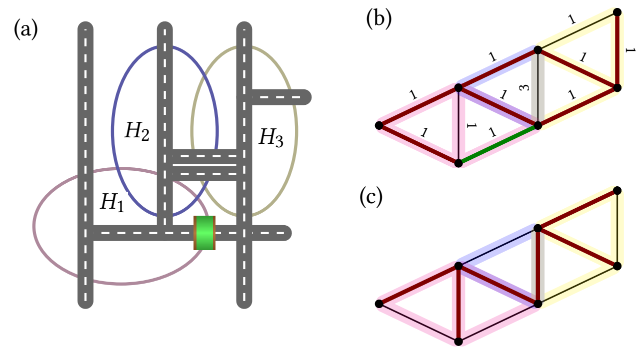

Figure 1: (a) A road network fragmenting habitats H_1, H_2, and H_3. (b & c) The corresponding graph and a solution (thicker edges) for (b) d-Diam GBP-C of minimum cost where the bridge over the broader road has cost 3 (while other bridges have cost 1) and the already built green bridge forms a forced edge; (c) 2-Diam GBP of minimum cardinality where edge costs and forced edges are ignored.

Fluschnik and Kellerhals [17] developed a framework to tackle such questions. They use a graph in which the vertices represent patches of land, and edges correspond to places where wildlife crossings can be installed. The *habitats* of animals then correspond to vertex subsets. Fluschnik and Kellerhals [17] define multiple classes of problems that represent different demands of habitat connectivity. In this work, we focus on one of them.

Problem 1. d-Diameter Green Bridges Placement with Costs (d-Diam GBP-C)

Input: An undirected graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{N}_0$, a set $\mathcal{H} = \{H_1, \ldots, H_r\}$ of habitats with $H_i \subseteq V$ and $|H_i| \geq 2$ for all $i \in \{1, \ldots, r\}$, a set $F^* \subseteq E$ of *forced* edges, an integer $k \in \mathbb{N}_0$.

Question: Is there an edge subset $F \subseteq E$ with $F^* \subseteq F$ and $c(F) := \sum_{e \in F} c(e) \leq k$ such that in the graph $G' := (V, F)$, $\operatorname{diam}(G'[H]) \leq d$ for every $H \in \mathcal{H}$?

If all edges have *unit* cost 1, i.e., if $c \equiv 1$, then we call the problem short d-Diameter Green Bridges Placement (d-Diam GBP). Fluschnik and Kellerhals [17] study only d-Diam GBP and without forced edges. Thus, we generalize their model by allowing general edge costs. The (possibly empty) set of forced edges that must be contained in every solution only extends but does not generalize

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/OYZM4458

their model since any forced edge can be modeled by a habitat of size two. A forced edge models the case where a green bridge is already built or planned, and thereby allows for incremental approaches of reconnecting habitats in an area.

Example 1. Consider the example given in Figure 1 for $d = 2$. In (a), we are given three habitats H_1, H_2, and H_3 of different species fragmented by a street network. The middle horizontal street part is broader than the others, and a green bridge is already built over a part of the horizontal street at the bottom. In (b) we show the graph G obtained from (a) with a vertex for each inhabited land patch and with edge costs c (above each edge), where all edges have cost 1 except for the one that corresponds to the broader street (which has cost 3). Each habitat H is depicted by a subgraph $G[H]$ colored in correspondence with (a). The forced edge is drawn in green. Thick lines correspond to a solution of cost 8, which is minimum. In (c), we show the graph from (b) when edge costs and forced edges are ignored. Thick lines correspond to a minimum-cardinality solution. If the true costs were applied and the forced edges were added, the obtained solution would have cost 9. ◇

This work contributes to a better understanding of d-Diam GBP from an algorithmic and complexity-theoretic point of view. As a starting point, and to garner a better understanding of the structural properties of the problem, we focus solely on the case $d = 2$. There are multiple reasons for this restriction: First, as the problem is trivial for $d = 1$ [17], we focus on the first value of d for which the problem is NP-hard. Second, the case $d = 2$ has also been studied the most in related problems (see below). Lastly, $d = 2$ is also motivated by a practical point of view: The way human infrastructure fragments the landscape often does not allow to reconnect habitats with a diameter of 1 (a clique). A good example is a habitat that contains the crossing of two streets: Such a habitat will induce a cycle on four vertices, which already has diameter 2.

Our main interest is the complexity of 2-Diam GBP-C on *planar* graphs as well as graphs with bounded *maximum degree*. As in our setting, the graphs correspond to the duals of road networks, these are natural assumptions. Moreover, we focus on instances in which the habitats are small. Small habitats appear more often for small mammals, amphibians, and reptiles [21, 22].

Related Work. The problems introduced by Fluschnik and Kellerhals [17] are not the first *network design problems* for *wildlife preservation* studied from a theoretical and algorithmic perspective. For designing wildlife corridors, Lai et al. [32] consider the problem of finding a subgraph induced by a vertex set so that within each habitat, a set of habitat-specific terminals is connected. LeBras et al. [33] extend this problem to guarantee multiple disjoint paths between each terminal pair. Both studies highlight the importance of focusing on planar inputs.

The need to "connect fragmented habitats" also appears in a range of areas outside of wildlife preservation. Applications include social networks [3], graph drawing [7, 8, 13, 29], combinatorial auctions [11], vacuum technology [14], structural biology [2], and reconfigurable computer networks [10, 16]. In the latter context, Chockler et al. [10] call for algorithms that "connect habitats" with a small diameter. Chen et al. [9] list further applications of the studied problem and its variants. Another problem concerned with connecting more than one habitat is Steiner Forest [4], extending the well-known Steiner Tree problem.

Table 1: Overview on our results for 2-Diam GBP regarding maximum degree Δ, maximum habitat size, and further restrictions like planarity. Recall that n, m, and r denote the number of vertices, of edges, and of habitats, respectively. (*: on planar graphs; †: even for unit costs)

<table>
<tr><th rowspan="2">Δ</th><th colspan="3">Maximum habitat size</th></tr>
<tr><th>≤ 3</th><th>= 4</th><th>≥ 5</th></tr>
<tr><td>≤ 3</td><td colspan="3">$O(n + r)$ [Thm. 2]</td></tr>
<tr><td>= 4</td><td colspan="2">$O(n + r)$ [Thm. 3]</td><td>open</td></tr>
<tr><td>≥ 5</td><td colspan="3">NP-hard† [Thm. 4(i)]</td></tr>
<tr><td>≥ 5*</td><td>$O(n^2r^2 + r^3)$ [Thm. 1]</td><td colspan="2">NP-hard† [Thm. 4(ii)]</td></tr>
</table>

Green bridges placement. Fluschnik and Kellerhals [17] proved that 2-Diam GBP is NP-hard even if $r = 1$, and polynomial-time solvable if the graph has constant maximum degree and there are only a constant number of habitats. Herrendorf et al. [24] show that an algorithm solving the problem in $2^{o(n^2+r)}$ time would break the exponential time hypothesis, even if each habitat has size 4. Several related problems consider the task of connecting just one "habitat". Minimizing the diameter of the habitat with a fixed budget is also NP-hard [37], as is the problem of adding few edges to a graph to reduce its diameter [18, 38]. The latter problem is also known as *finding shortcut edges* and has been studied intensely [1, 6, 12].

The variant where every habitat has size three has been studied extensively and is NP-hard even if G is a clique [16, 23, 25]. It is 2-approximable in polynomial time, which is best possible under the unique games conjecture [25]. The variant is polynomial-time solvable on planar graphs if each habitat induces a triangle being the boundary of a face, or when the graph has maximum degree and each habitat induces a triangle [23]. If there is a solution that spans a tree in G, then it can be found in linear time [30].

2-Diam GBP has been studied by Jansson et al. [28] who give a polynomial-time $O(\eta^4)$-approximation algorithm, where η denotes the maximum habitat size. Moreover, there are several studies on heuristics for 2-Diam GBP [34–36]. Besides the primary objective, the heuristics aim to keep the maximum degree of $G[F]$ low.

Another related network design problem is network sparsification. Gionis et al. [20] consider a variant of 2-Diam GBP called Sparse Stars where an edge subset F is a solution if for each $H \in \mathcal{H}$ the graph $G[F][H]$ contains a spanning star. As a star has diameter two, any such F is also a solution for 2-Diam GBP. They give a polynomial-time approximation algorithm for Sparse Stars. Herrendorf et al. [24] study the parameterized complexity of Sparse Stars with respect to the parameters solution size $|F|$ and maximum degree Δ, among others. Korach and Stern [31] examine a variant of Sparse Stars which additionally requires that the solution F induces a spanning tree of G.

Our Contributions. Our results are summarized in Table 1. We provide tractability boundaries (i.e., polynomial-time solvability versus NP-hardness) for 2-Diam GBP-C when combining sparsity as described above with small maximum habitat sizes. On the positive

side, we show that 2-Diam GBP-C is polynomial-time solvable for instances with
(i) maximum degree three;
(ii) maximum degree four and maximum habitat size four;
(iii) planar graphs and maximum habitat size three.

Cases (i) and (ii) are even solvable in linear time. To complement these results, we show that 2-Diam GBP is NP-hard even if the input graph has maximum degree five and
(i) the maximum habitat size is three;
(ii) is also planar, and the maximum habitat size is four.

For all of our polynomial-time algorithms, the habitat size is constant. Therefore, the complexity of these specific cases arises from the way the habitats intersect. For each of the polynomial-time solvable cases, we make use of different structural observations and algorithmic techniques.

Highlights of this work are the structural observations and polynomial-time algorithms for instances with planar graphs and habitat size three (Section 3) and instances with maximum degree four and habitat size four (Section 5). The former algorithm exploits that size-three habitats in planar graphs admit a hierarchy which we can break down with an intricate data reduction rule. For the latter result, we introduce the habitat intersection graph that captures the complexity of how habitats intersect; in this case, we are able to show that each connected component of the intersection graph is a path or a cycle, or has constant size. For larger habitat sizes, we were unable to prove the same structure for the habitat intersection graph, but we conjecture that it keeps the same structure. Both structural properties cease to hold when increasing the habitat size to four in planar graphs, or the maximum degree to five, even for size-three habitats. This is reflected by our hardness results.

We first define some notation and state some preprocessing rules used throughout the paper in Section 2. Then, we present the algorithms, starting with the algorithm on planar graphs in Section 3. Afterwards, our results are ordered by increasing maximum degree (Sections 4 & 5), ending with the hardness results for graphs of maximum degree five (Section 6).

2 PRELIMINARIES AND PREPROCESSING

For an (undirected) graph $G = (V, E)$, we also denote by $V(G)$ and $E(G)$ the vertex and edge sets and let $n := |V|$ and $m := |E|$. For a vertex set $W \subseteq V$, the graph $G[W]$ *induced by* W is the graph with vertex set W and all edges in E with both endpoints in W. For an edge set $E' \subseteq E$, the graph *induced by* E' is $G[E'] := (V, E')$. For an edge set E' and a vertex set $W \subseteq V$, the graph $G[E'][W]$ is the graph $G[E']$ induced by W. For $u, v \in V$, let $\operatorname{dist}_G(u, v)$ be the length of a shortest path between u and v. The *diameter* of G is $\operatorname{diam}(G) := \max_{u,v \in V} \operatorname{dist}_G(u, v)$. A vertex v is called a *cut vertex* if $G[V \setminus \{v\}]$ contains more connected components than G. The degree $\deg_G(v)$ of a vertex v is the number of edges incident to v. The maximum degree of G is the maximum degree of a vertex in G. A graph is k-connected if for each pair u, v of vertices, there are at least k vertex-disjoint paths connecting them. A graph is planar if it can be drawn in the 2-dimensional plane without crossing edges. A set $W \subseteq V$ is a vertex cover of G if $G[V \setminus W]$ is edgeless.

Some proofs and details (marked by ★) are deferred to the full version of the paper [40].

Preprocessing. We use the following straightforward reduction rules throughout our paper.

Reduction Rule 1. *Return* no *if* $\operatorname{diam}(G[H]) > 2$ *for some* $H \in \mathcal{H}$.

If there is an edge $e = \{u, v\}$ contained in a habitat H such that the distance from u to v is more than two when removing the edge, then e must be in every solution. Thus:

Reduction Rule 2. *If there is a habitat $H \in \mathcal{H}$ and an edge $e \in E(G[H])$ not contained in a triangle in $G[H]$, then mark e as forced.*

Reduction Rule 2 implies the next more direct rule.

Reduction Rule 3. *If a habitat $H \in \mathcal{H}$ contains no triangle, then mark each edge in $G[H]$ as forced. If $G[H]$ for some $H \in \mathcal{H}$ contains a cut vertex v, then mark all edges in $G[H]$ incident to v as forced.*

Next, we remove all unnecessary habitats, edges, and vertices, and small components.

Reduction Rule 4. *If $\operatorname{diam}(G[F^*][H]) \leq 2$ for a habitat $H \in \mathcal{H}$, then remove H from $\mathcal{H}$.*

Reduction Rule 5. *If there is an edge $e \in E$ such that there is no habitat $H \in \mathcal{H}$ with $e \in E(G[H])$, then remove e. If further $e \in F^*$, then also set $k := k - c(e)$.*

Reduction Rule 6. *If $v \in V$ is isolated, then remove v.*

Reduction Rule 7. *Let C be a connected component of G with $|V(C)| \leq 6$. Then, in constant time, brute-force a minimum-cost set $F^*_C \subseteq E(C)$ with $F^* \cap E(C) \subseteq F^*_C$ such that $\operatorname{diam}(G[F^*_C][H]) \leq 2$ for every habitat $H \subseteq V(C)$. Next, remove C from G and all habitats $H \subseteq V(C)$ from $\mathcal{H}$, and set $k := k - c(F^*_C)$.*

Observation 1 (★). *Reduction Rules 1 to 7 can be exhaustively applied to an instance $(G, c, \mathcal{H}, F^*, k)$ of* 2-Diam GBP-C *in $O(r \cdot \eta^3 + n + m)$ time, where $\eta := \max_{H \in \mathcal{H}} |H|$. In the obtained instance, each component has more than 6 vertices and for each habitat H, $G[H]$ (i) is 2-connected, (ii) has at least two unforced edges, and (iii) at least one triangle.*

We mention in passing that, after applying our preprocessing, habitats in instances with maximum degree 3 (4) have size at most 6 (8). This can be verified by checking all graphs with maximum degree 4 and diameter 2, which contain at most 15 vertices [15].

3 PLANAR GRAPHS AND HABITATS OF SIZE THREE

Herkenrath et al. [23] showed that, when every habitat has size three and induces a face in a planar graph, 2-Diam GBP-C is efficiently solvable.[1] We extend their result by showing that this is possible without the face constraint.

Theorem 1 (★). 2-Diam GBP-C *on planar graphs is solvable in $O(n^2 r^2 + r^3)$ time if the habitat size is at most three.*

We assume that the graph is embedded in the plane $\mathbb{R}^2$ using a straight-line embedding ϕ and that each habitat induces a triangle by Observation 1. The crucial observation is that for any habitat

[1]They show this for a variation of 2-Diam GBP-C where one only needs that $G[F][H]$ is *connected* for each $H \in \mathcal{H}$. For triangle habitats, this is equivalent to our diameter constraint.

$H \in \mathcal{H}$ of size 3, $G[H]$ splits the plane into an inner (bounded) area $A_H^{\text{in}} \subseteq \mathbb{R}^2$ and an outer, unbounded area $A_H^{\text{out}} \subseteq \mathbb{R}^2$. A vertex or edge is contained in (i) $G[H]$, (ii), the inner, or (iii) the outer area. Habitats cannot be both inside and outside as otherwise two edges would cross. This implies a hierarchy on the habitats: For every two habitats, either one has a vertex in the inner area of the other, or both habitats consider the other as outside of them. Hence, there are habitats that induce faces (at the lowest level of the hierarchy). Thus, there is a habitat that contains only face habitats; we call such a habitat *reducible*, see Definition 2.

As habitats inside a reducible habitat H may share edges with H, the minimum cost of reconnecting them depends on the subsolution, i.e., the edges in $G[F][H]$. The main tool for Theorem 1 is Reduction Rule 8, which eliminates all vertices, edges, and habitats inside H, propagating the costs of its optimal inside solutions into the cost of the edges of H. This is remarkable since there are four possible subsolutions for H, but only three edges to store the information in: A solution may contain all or all but one of the edges in H.

Exhaustive application of our reduction rule results in an instance in which every habitat induces a face and allows us to apply the algorithm by Herkenrath et al. [23].

Let us introduce some notation. For a habitat $H \in \mathcal{H}$, a vertex $v \in V \setminus H$ is either inside or outside a habitat H, i.e., $\phi(v) \in A_H^{\text{in}}$ or $\phi(v) \in A_H^{\text{out}}$ (we define the vertices in H to be neither inside nor outside of H). As G is planar, the endpoints of an edge in $E \setminus E(G[H])$ cannot be both inside and outside H. Hence, $e \in E \setminus E(G[H])$ is inside if e has an endpoint in A_H^{in} and outside if e has an endpoint in A_H^{out}. Finally, we define a habitat $H' \in \mathcal{H}$, $H' \neq H$, to be inside if $A_{H'}^{\text{in}} \subsetneq A_H^{\text{in}}$ and outside if $A_{H'}^{\text{out}} \subsetneq A_H^{\text{out}}$. We denote by $V_H^{\text{in}}, V_H^{\text{out}}, E_H^{\text{in}}, E_H^{\text{out}}, \mathcal{H}_H^{\text{in}}, \mathcal{H}_H^{\text{out}}$, the sets of vertices, edges, and habitats that are inside or outside of H. Note that H induces an (inner) face if and only if $\mathcal{H}_H^{\text{in}} = \emptyset$.

Definition 2. We call a habitat $H \in \mathcal{H}$ *reducible* if $\mathcal{H}_H^{\text{in}} \neq \emptyset$ and for each $H' \in \mathcal{H}_H^{\text{in}}$ it holds true that $\mathcal{H}_{H'}^{\text{in}} = \emptyset$.

We prove that if at least one habitat contains another, then there is a reducible habitat.

Lemma 3 (★). *Let G be a non-empty finite planar graph with an embedding into the plane and $\mathcal{H} \neq \emptyset$ a set of pairwise distinct habitats in G each inducing a triangle. Then either $\mathcal{H}_H^{\text{in}} = \emptyset$ for all $H \in \mathcal{H}$ or there is a reducible habitat.*

We next define some terms that we need for Reduction Rule 8, alongside an example in Figure 2. Let $H \in \mathcal{H}$ be a habitat; define $E_H := E(G[H])$. The *inner graph* G_H *of* H is defined as the subgraph of G containing all edges in $E_H^{\text{in}} \cup E_H$. The *inner cost function* c_H assigns cost $c_H(e) := c(e)$ to each $e \in E_H^{\text{in}}$ and cost $c_H(e) := 0$ to each $e \in E_H$. We define the *inside optimum solution* F_H for H to be an edge set $E(G_H) \cap F^* \subseteq F_H \subseteq E(G_H)$ satisfying $H' \subseteq V(G_H[F_H])$ and $\operatorname{diam}(G[F_H][H']) \leq 2$ for all $H' \in \mathcal{H}_H^{\text{in}}$ that minimizes the cost $c_H(F_H) =: \operatorname{opt}_H$. Intuitively, F_H is the cheapest solution regarding the inner cost function that contains all forced edges satisfying the diameter constraints for all habitats inside of H. In the example in Figure 2, $F_H = E_H \cup \{v_r, v_c\}$ and $\operatorname{opt}_H = 3$.

For $e \in E_H$, the *e-omitting inner cost function* c_H^e sets $c_H^e(e) := 1 + \sum_{e'' \in E_H^{\text{in}}} c(e'')$ and $c_H^e(e') := 0$ for each $e' \in E(G_H) \setminus \{e\}$. The *$e$-omitting inside optimum solution* F_H^e and $\operatorname{opt}_H^e := c_H^e(F_H^e)$ are

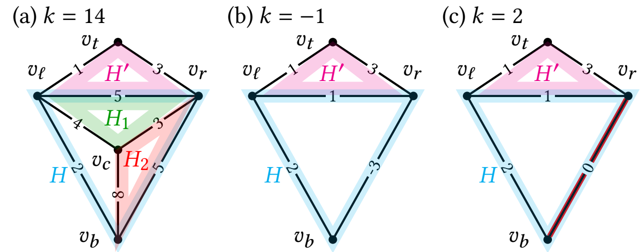

Figure 2: An illustration of the application of Reduction Rule 8, with edge costs written on the edges and k given in the subfigure's label. (a) The habitats are $H' = \{v_\ell, v_r, v_t\}$, $H = \{v_\ell, v_r, v_b\}$, $H_1 = \{v_\ell, v_r, v_c\}$, and $H_2 = \{v_r, v_b, v_c\}$, where H is the reducible habitat with H_1 and H_2 being inside H. (b) After deleting H_1 and H_2 and adjusting edge costs and k. (c) After dealing with negative edge costs, where edge $\{v_r, v_b\}$ (thick, red) is now forced.

defined just as the inside optimum solution with the only difference being the cost function. In the example in Figure 2, $F_H^{\{v_r, v_b\}} = F_H \setminus \{\{v_r, v_b\}\} \cup \{\{v_\ell, v_c\}\}$ and $\operatorname{opt}_H^{\{v_r, v_b\}} = 11$. Further, $\operatorname{opt}_H^{\{v_\ell, v_b\}} = 3$ and $\operatorname{opt}_H^{\{v_\ell, v_r\}} = 7$.

Since the cost of e is prohibitively high, we have $e \notin F_H^e$. Thus $\operatorname{opt}_H \leq \operatorname{opt}_H^e \leq \sum_{e' \in E_H^{\text{in}}} c(e')$, and observe the following.

Observation 4. *For $H \in \mathcal{H}$, $e \in E_H$, and a solution F,*

$$c(F \cap E_H^{\text{in}}) \geq \begin{cases} \operatorname{opt}_H, & \text{if } |F \cap E_H| = 3, \\ \operatorname{opt}_H^e, & \text{if } E_H \setminus F = \{e\}. \end{cases}$$

This gives rise to our central reduction rule (see Figure 2).

Reduction Rule 8 (★). *Let H be a reducible habitat. Then remove E_H^{in} from the graph, $\mathcal{H}_H^{\text{in}}$ from the habitat set, increase k by $2\operatorname{opt}_H - \sum_{e \in E_H \setminus F^*} \operatorname{opt}_H^e$, and increase the cost of each $e \in E_H$ by opt_H. Afterwards, decrease the cost of each $e \in E_H \setminus F^*$ by opt_H^e. For every edge e whose cost $c(e)$ becomes negative in this process, increase k by $|c(e)|$, set $c(e) := 0$, and mark e as forced.*

Proof idea. Denote by $\mathcal{I}$ and $\mathcal{I}'$ the instances before and after applying the rule. Let c' denote the new cost function. For the proof idea, we assume that $F^* = \emptyset$ and that no edges become negative during the application.

Consider a solution F of cost at most k for $\mathcal{I}$. Then either $|F \cap E_H| = 3$ or $E_H \setminus F = \{e\}$. Let $F' := F \setminus E_H^{\text{in}}$. The cost of the edges in $E_H \cap F'$ increases by $3\operatorname{opt}_H - \sum_{e' \in E_H} \operatorname{opt}_H^{e'}$ if $|F \cap E_H| = 3$ and by $2\operatorname{opt}_H - \sum_{e' \in E_H} \operatorname{opt}_H^{e'} + \operatorname{opt}_H^e$ if $E_H \setminus F = \{e\}$, but F' need not pay for the edges in E_H^{in}, saving $c(F \cap E_H^{\text{in}})$. So the cost increase $c'(F') - c(F)$ is equal to the budget increase $k' - k$.

For the reverse direction, if F' is a solution for $\mathcal{I}'$, we consider the same cases. If $|F' \cap E_H| = 3$, then we construct F by adding an inside optimum solution to F'. If $E_H \setminus F' = \{e\}$, then we add an e-omitting inside optimum solution. The verification that $c(F) \leq k$ is the same as above.

It remains to show that the constructed solutions satisfy the habitat diameter constraints and how to deal with forced and negative edges. These details are in the paper's full version [40]. □

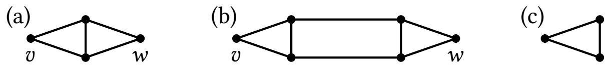

Figure 3: The zones, where the K_3 in (c) must not be contained in (a) or (b).

Algorithm 1: Given a reduced instance $I = (G = (V, E), c, \mathcal{H}, F^*, k)$ where $\Delta(G) \leq 3$, return a set $F^* \subseteq F \subseteq E$ or no. Here, $E_{H-X} \coloneqq E(G[H]) \setminus E(X)$ and $F_{H,X} \coloneqq E_{H-X} \cup F_X$.

Initialize $F \leftarrow F^*$;
foreach *zone* $X \subseteq G$ **do**
  Find lowest-cost set $F^* \cap E(X) \subseteq F_X \subseteq E(X)$ such that $\mathrm{diam}(G[F_{H,X}][H]) \leq 2$ for each H with $H \cap V(X) \neq \emptyset$;
  $F \leftarrow F \cup F_X$;
if $c(F) \leq k$ **then return** F **else return** no;

When applying Reduction Rule 8 to a reducible habitat, its parent becomes reducible. Thus, after exhaustive application, every habitat induces a face, and we can apply Herkenrath et al. [23]'s algorithm. This proves Theorem 1. See the paper's full version [40] for details.

4 MAXIMUM DEGREE THREE

We next give a linear-time algorithm for subcubic graphs.

Theorem 2 (★). 2-Diam GBP-C *is solvable in $O(n + r)$ time on graphs of maximum degree three.*

Due to Reduction Rule 2, we only care about those edges of habitats contained in a triangle, and every edge of a habitat not contained in a triangle is forced. Indeed, the degree and diameter constraints allow us to partition the triangles into vertex-disjoint groups each of which contains at most two triangles. We call these groups *zones*, see Figure 3. The clue now is that any edge of a habitat outside of a zone is forced. As the zones have constant size, we can brute-force an optimum solution for each. The union of these solutions then is an optimum solution; see Algorithm 1.

Let $M(K_3)$ be the graph obtained by adding a complete matching between two K_3's. Observe that, due to the degree bound, any K_4 and $M(K_3)$ would be isolated in G and thus removed by Reduction Rule 7. Let $K_4 - e$ be a K_4 where one edge is removed. Let $M(K_3) - e_M$ be a $M(K_3)$ where one matching edge is removed.

Definition 5. A *zone* is a subgraph of G isomorphic to a $K_4 - e$ or an $M(K_3) - e_M$, or a K_3 that is not contained in a $K_4 - e$ or $M(K_3) - e_M$; compare with Figure 3.

We next show that zones are vertex-disjoint.

Observation 6. *Any two distinct zones are vertex-disjoint.*

Proof. In a graph with maximum degree three, any two distinct triangles either share an edge or are vertex-disjoint. If they share two edges, then they are not distinct. If they share no edge but a vertex, then this vertex has degree four. Moreover, a triangle can share edges with at most one other triangle, otherwise one of its vertices would have degree four in G. This implies that the $K_4 - e$ shares no vertices or edges with other zones. Finally, as the only two vertices in the $M(K_3) - e_M$ with degree less than three are not adjacent, it cannot share a vertex or edge with any other zone. □

Observation 7. *Let $H \in \mathcal{H}$ and $v, w \in H$. Then all v-w-paths of length at most two in $G[H]$ contain edges of at most one zone.*

Proof. Note that there cannot be a single such path containing edges of two zones as this would contradict their vertex-disjointness. Suppose towards a contradiction that there are two distinct zones X and Y such that $G[H]$ contains two length-two v-w-paths $P_X = (v, x, w)$ and $P_Y = (v, y, w)$ such that P_X contains an edge from X and P_Y contains an edge from Y. Assume that $v, x \in X$ and $w, y \in Y$. Note that v, x, w, y induce a C_4: $\{v, w\}$ cannot exist due to zones being vertex-disjoint and $\{x, y\}$ cannot exist as otherwise the vertices would have degree greater than three (note that each of x, y has at least one further neighbor in their respective zone). Observe further that $\deg_X(v), \deg_X(x) \leq 2$ and v and x are adjacent. This is only possible if X is a triangle. Analogously, Y is also a triangle. But then, $G[X \cup Y]$ induces an $M(K_3) - e_M$, contradicting the fact that X and Y are distinct zones. □

Proof of Theorem 2. We first prove that F is a solution and then that it has minimal cost. Finally, we show that Algorithm 1 runs in $O(n + r)$ time.

To see that F is a solution, we show $\mathrm{dist}_{G[F][H]}(v, w) \leq 2$ for each $H \in \mathcal{H}$ and each $v, w \in H$. By Observation 7, all v-w-paths of length two contain edges from at most one zone. If they contain edges of zero zones, then they are all forced and $\mathrm{dist}_{G[F][H]}(v, w) \leq 2$. If they contain edges of zone X, then all edges outside of that zone are forced; thus $\mathrm{dist}_{G[F][H]}(v, w) = \mathrm{dist}_{G[F_{H,X}][H]}(v, w) \leq 2$.

Now suppose that there is a solution F' with $c(F') < c(F)$. Since F and F' can only differ in edges contained in triangles and every triangle is contained in a zone, there is a zone X with $c(E(X) \cap F') < c(E(X) \cap F)$. This contradicts the choice of F_X by Algorithm 1.

Finally, for the running time, note that every habitat has constant size due to the degree and diameter constraints, or we have a trivial no-instance. Thus, Reduction Rules 1 to 7 are applicable in $O(n + r)$ time. Since all zones are vertex-disjoint, there are at most n zones, which can be found in linear time by searching the neighborhoods of each vertex. Finally, for the brute-force step of the algorithm, observe that zones and habitats have constant size and that each zone contains only a constant number of habitats. □

5 MAXIMUM DEGREE FOUR

In this section, we will show the following result.

Theorem 3 (★). 2-Diam GBP-C *is solvable in $O(n + r)$ time if the maximum degree of G and the maximum habitat size are at most 4.*

Note that, even if the habitats have size at most four, we cannot remove habitats that are subsets of other habitats: Consider a habitat $H = \{u, v, w, x\}$ that induces a K_4. Then a solution F with $\{u, v\}, \{u, w\}, \{u, x\} \in F$ (but no other edges in $G[H]$) is sufficient for H; however, for the subset habitat $H' = \{v, w, x\}$, F would be infeasible. Our following algorithm cannot deal with such habitats. Instead, we simply compute the set $\mathcal{F}_H$ of feasible solutions for each habitat $H \in \mathcal{H}$; note that each set $\mathcal{F}_H$ contains only constantly many solutions as H is of constant size. If we then have a habitat H' which is a subset of some habitats $H \in \mathcal{H}'$, then we remove H'

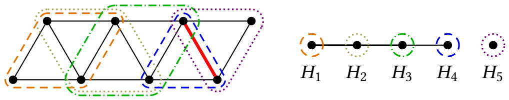

Figure 4: Illustration of a graph with five habitats, and the corresponding intersection graph $\partial\mathcal{H}$ by Definition 9. Note that H_1 and H_3 are not neighboring in $\partial\mathcal{H}$ as $E(G[H_1 \cap H_3]) \subseteq E(G[H_1 \cap H_2])$, and that H_4 and H_5 are not neighboring as their only common edge is forced (red).

and keep only those solutions in $\mathcal{F}_H$, $H \in \mathcal{H}'$ that are feasible for both H and H'.

We assume that Reduction Rules 1 to 7 are inapplicable, implying that each habitat induces a K_3, a K_4, or a $K_4 - e$. Formally, we reduce 2-Diam GBP-C to the following problem.

Problem 2. General Green Bridges Placement with Costs (Gen GBP-C)

Input: An undirected graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{N}_0$, a set $\mathcal{H} = \{H_1, \ldots, H_r\}$ of habitats with $H_i \subseteq V$ and $|H_i| \geq 2$ for all $i \in \{1, \ldots, r\}$, a set $\mathcal{F}_{H_i} \subseteq 2^{E(G[H_i])}$ of feasible edge sets for all $i \in \{1, \ldots, r\}$, a set $F^* \subseteq E$ of *forced* edges, and an integer $k \in \mathbb{N}_0$.

Question: Is there a subset $F \subseteq E$ with $F^* \subseteq F$ and $c(F) := \sum_{e \in F} c(e) \leq k$ such that for every $i \in \{1, \ldots, r\}$ there is a set $F_{H_i} \in \mathcal{F}_{H_i}$ with $F_{H_i} \subseteq F$?

Observation 8. *For each instance $\mathcal{I} = (G, c, \mathcal{H}, F^*, k)$ of* 2-Diam GBP-C *with constant habitat size, one can compute in $O(r)$ time the sets $\mathcal{F}_H \subseteq 2^{E(G[H])}$, $H \in \mathcal{H}$, such that F is a solution for $\mathcal{I}$ if and only if F is a solution for the instance $\mathcal{I}' = (G, c, \mathcal{H}, \{\mathcal{F}_H\}_{H \in \mathcal{H}}, F^*, k)$ of* Gen GBP-C.

We apply the following rule to Gen GBP-C instances.

Reduction Rule 9. *Let $H' \in \mathcal{H}$ and let $\mathcal{H}' := \{H \in \mathcal{H} \mid H' \subseteq H\}$. Then, for each $H \in \mathcal{H}'$, keep only those sets F_H in $\mathcal{F}_H$ such that there is a set $F_{H'} \in \mathcal{F}_{H'}$ with $F_{H'} \subseteq F_H$, and then remove H' and $\mathcal{F}_{H'}$.*

5.1 The Habitat Intersection Graph

The crucial part for computing a solution for Gen GBP-C (and thus also 2-Diam GBP) is that a habitat may influence the subsolution for another habitat (directly) only if they both share an (unforced) edge. However, if the two habitats have no common edges, but both share edges with a third habitat, then their subsolutions can still affect each other. To exploit this property, we formally define the habitat intersection graph, where two habitats are neighboring—up to one small caveat—when they share an unforced edge (see Figure 4 for an illustrative example).

Definition 9. For a graph G with a subset of edges being forced and habitat set $\mathcal{H}$, the *intersection graph* $\partial\mathcal{H}$ has the vertex set $V(\partial\mathcal{H}) = \mathcal{H}$ and two habitats are *neighboring*, i.e., $\{H_1, H_2\} \in E(\partial\mathcal{H})$ if and only if

(i) $G[H_1 \cap H_2]$ contains at least one unforced edge and
(ii) there is no $H_3 \in \mathcal{H}$ such that $E(G[H_1 \cap H_2])$ is a proper subset of $E(G[H_1 \cap H_3])$ *or* $E(G[H_2 \cap H_3])$.

We call the edges in $G[H_1 \cap H_2]$ *intersection edges* between H_1 and H_2. For a subset $C \subseteq \mathcal{H}$, we denote by $V(C)$ and $E(C)$ the vertices and the edges of $G[\bigcup_{H \in C} H]$.

We show that we can compute $\partial\mathcal{H}$ efficiently and solutions separately for each of its connected components.

Observation 10 (★). *If the maximum habitat size and the maximum degree of the graph are both at most four, then each edge is contained in at most 21 habitats.*

Lemma 11. *Given a graph G with maximum degree four, a subset of forced edges, and a set $\mathcal{H}$ of r habitats each of size at most four, one can compute $\partial\mathcal{H}$ in $O(n + r)$ time.*

Proof. We first compute for each edge e the set of habitats containing e (which is constant by Observation 10) in two steps: First, compute for each vertex v the set of habitats containing v (which is also constant by Observation 10), then perform a breadth-first search over G and, whenever there is an edge $e = \{u, v\}$ with $u, v \in H$, add H to the habitat set for e. Next, for every unforced edge e and every pair H_1, H_2 of habitats containing e, we check Definition 9(ii) by listing all habitats sharing an edge with H_1 or H_2 in constant time. If there is no $H_3 \in \mathcal{H}$ with $E(G[H_1 \cap H_2]) \subseteq E(G[H_i \cap H_3])$ for $i \in \{1, 2\}$, then add $\{H_1, H_2\}$ to $E(\partial\mathcal{H})$. As there are $O(n)$ edges, the claimed running time follows. □

Lemma 12. *For an instance of* Gen GBP-C, *let $C_1, C_2, \ldots, C_\ell$ be the connected components of $\partial\mathcal{H}$. For each $i \in \{1, \ldots, \ell\}$, define $G_i := G[\bigcup C_i]$, and let $F_i \subseteq E(G_i)$ be a minimum-cost solution for the subgraph G_i. Then $F := F^* \cup F_1 \cup \cdots \cup F_\ell$ is a minimum-cost solution for G, where F^* is the set of forced edges.*

Proof. If $G[H_1 \cap H_2]$ contains edges but $\{H_1, H_2\} \notin E(\partial\mathcal{H})$, then all intersection edges are in F^*, or there is a habitat H_3 as in Definition 9(ii). In the latter case, $\{H_1, H_3\}, \{H_3, H_2\} \in E(\partial\mathcal{H})$; thus H_1, H_2, H_3 are in the same connected component. Thus, for every $i, j \in \{1, \ldots, \ell\}$, we have $E(G_i) \cap E(G_j) \subseteq F^*$. Consequently, $F_i \cap F_j \subseteq F^*$. Clearly, F is a solution, and every solution must contain F^*. As every F_i is cost-optimal for G_i, the resulting solution F is also cost-optimal. □

5.2 Structure of the Habitat Intersection Graph

The following important vertices limit how habitats intersect.

Definition 13. For two neighboring habitats $H, H' \in \mathcal{H}$, we call a vertex $v \in H \cap H'$ *docking towards* H' if it is adjacent to a vertex in $H' \setminus H$.

Observe that a vertex $v \in H \cap H'$ may be docking towards H', but not towards H; i.e., docking vertices are not symmetric. Also, there are always at least two docking vertices towards H' in $H \cap H'$ if H and H' are neighboring.

Observation 14 (★). *Let $H, H' \in \mathcal{H}$ be two neighboring habitats. Then there are at least two and at most* $\max\{|H|, |H'|\} - 1$ *vertices in H docking towards H'.*

Note that the docking vertices towards H' may have at most three neighbors within H, otherwise they cannot be adjacent to any vertex in $H' \setminus H$. Indeed, we show that among the intersecting edges, there must be one whose endpoints both have degree at most three in H. We call them *docking edges*, and additionally *proper* if one of their endpoints has degree two. We define these notions for sets of habitats.

Definition 15. For a set $C \subseteq \mathcal{H}$ of habitats, an edge $\{u, v\} \in E(C)$ is called *docking* if $\deg_C(u) \leq 3$ and $\deg_C(v) \leq 3$, and *properly docking* if $\deg_C(u) = 2$ and $\deg_C(v) = 3$.

Observation 16. *If a set $C \subseteq \mathcal{H}$ does not contain docking edges, then C has no neighboring habitats.*

Proof. Suppose that there is a habitat $H \notin C$ that is neighboring a habitat $H' \in C$. Then there are two vertices $u, v \in H \cap H'$ that are docking towards H, and there is one vertex $x \in H \setminus V(C)$. Thus, the degree of both of these vertices within C is at most three. So if $\{u, v\} \in E$, then C contains a docking edge. Hence, $\{u, v\} \notin E$, and as $H \cap H'$ contains an unforced edge, there is another vertex $w \in H \cap H'$, and $\{u, w\}, \{w, v\} \in E$. As C contains no docking edge, we have $\deg_C(w) = 4$, and so $\{w, x\} \notin E$. Thus, H induces a C_4 (or $G[H]$ contains even fewer edges); a contradiction to the inapplicability of Reduction Rule 2. ◻

We now show a monotonicity property when there are no proper docking edges in C.

Lemma 17. *Let $C \subseteq \mathcal{H}$ be a set of at least two habitats that are connected in $\partial\mathcal{H}$. Let $H \in \mathcal{H} \setminus C$ be adjacent to at least one habitat in C. If for each vertex $v \in V(C)$ docking towards H we have $\deg_C(v) \geq 3$, then $C' := C \cup \{H\}$ contains strictly fewer docking edges than C. Moreover, C' does not contain a docking edge with an endpoint in H.*

Proof. Suppose first that $H \cap V(C)$ contains exactly two vertices u, v. Then both are docking vertices, and $\deg_C(u) = \deg_C(v) = 3$. Moreover, $\{u, v\} \in E$, or H is not neighboring any habitat in C. If H contains four vertices, then the two (adjacent) vertices in $H \setminus V(C)$ both have degree two; thus H induces a C_4, which contradicts the inapplicability of Reduction Rule 2. If $H = \{u, v, w\}$, then $\deg_{C'}(w) = 3$ and $\deg_{C'}(u) = \deg_{C'}(v) = 4$. Thus, the new vertex $w \in V(C') \setminus V(C)$ is not incident to a docking edge, and $\{u, v\}$ is docking in C but not in C'. As the remaining edges are unaffected, there are strictly fewer docking edges in C'.

Now, suppose that $H \cap V(C)$ contains exactly three vertices u, v, w. Then H contains a fourth vertex $x \notin V(C)$. If one of the vertices in $H \cap V(C)$, say v, is not docking, then it must be adjacent to both u and w; otherwise $G[H]$ is not 2-connected. Then $\{v, x\} \notin E$ (otherwise v is docking); thus u, v, w form a triangle in G (otherwise H induces a C_4). Then none of the edges incident to $x \in V(C') \setminus V(C)$ are docking, but $\{u, w\}$ is docking in C' but not in C. As the remaining edges are unaffected, the number of docking edges in C' strictly decreases. Suppose next that v is docking. If u, v, w form a triangle in G (i.e., H induces a K_4), then we are in a case identical to the three-vertex case above. If one of the edges between u, v, w, say, $\{u, w\}$, is missing, then H induces a K_4-e. In this case, $\{u, v\}$ and $\{v, w\}$ are docking in C, but not in C'; and there is no docking edge incident to x as $\deg_{C'}(u) = \deg_{C'}(w) = 4$ and $\deg_{C'}(x) = 3$. Thus, the number of docking edges strictly decreases. ◻

Next we show that, if C contains proper docking edges, then C is part of a path in $\partial\mathcal{H}$.

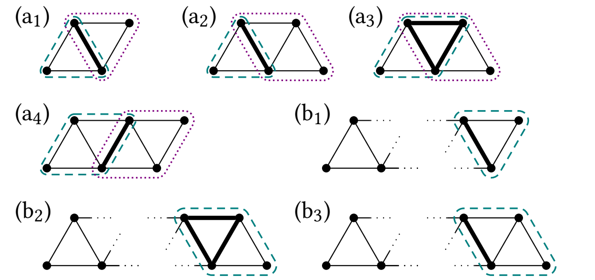

Figure 5: Illustration to Lemma 18 with base cases (a$_1$)–(a$_4$), with two habitats (dashed/dotted), and induction step cases (b$_1$)–(b$_3$) with habitat H (dashed). Thick edges are shared.

Lemma 18 (★). *Let $C \subseteq \mathcal{H}$ be a set of at least two habitats that are connected in $\partial\mathcal{H}$. If C contains at least one proper docking edge, then $\partial\mathcal{H}[C]$ is a path, and each habitat corresponding to an endpoint of the path contains exactly one degree-2 vertex with a degree-3 neighbor.*

Proof sketch. We proceed by induction on the number of habitats in C. Our base case is $|C| = 2$. Figure 5(a) shows habitat pairs containing at least one proper docking edge. Assume the statement holds for any set of ℓ habitats. Consider a set C' of $\ell + 1$ habitats such that $\partial\mathcal{H}[C']$ is connected and C' contains a habitat H with a proper docking edge. Let $C := C' \setminus \{H\}$ and assume $V(C) \neq V(C')$, otherwise we are done. In this sketch, we verify the ways that H neighbors C in Figure 5(b). Note that $H \cap V(C)$ contains a proper docking edge e of C; otherwise H does not contain a proper docking edge in C'. Moreover, C contains another degree-two vertex with a degree-three neighbor whose edge is disjoint from e. It remains to prove that C' is a path, i.e., that H has only one neighbor in C. For this, we need Definition 9(ii); details are in the full proof. ◻

From this, we can deduce the following.

Lemma 19. *Each component in $\partial\mathcal{H}$ is either a path or a cycle, or has constant size.*

Proof. We show that if there is a habitat H with at least three neighbors in $\partial\mathcal{H}$, then H is part of a constant-size component C'. Let C comprise H and its neighbors. By the contraposition of Lemma 18, C does not contain any proper docking edge. So any habitat added to C strictly decreases the number of docking edges of the subgraph by Lemma 17, and by Observation 16, C cannot grow without docking edges. As $|E(C)|$ initially is a constant, its number of docking edges is also constant; thus $|C'|$ is also constant. ◻

5.3 The Algorithm

We give an algorithm for Gen GBP-C where $\partial\mathcal{H}$ is a path.

Lemma 20 (★). *Let $\mathcal{I}$ be an instance of* Gen GBP-C *where $\partial\mathcal{H}$ is a path and Reduction Rule 9 has been exhaustively applied. If $\partial\mathcal{H}$ is given, then one can compute in $O(r \cdot \lambda^4)$ time a minimum-cost solution for $\mathcal{I}$. Here, $\lambda := \max_{H \in \mathcal{H}} |\mathcal{F}_H|$.*

Proof sketch. Let $(H_1, H_2, \ldots, H_r)$ be the path in $\partial\mathcal{H}$. Assume, without loss of generality, that for each $H \in \mathcal{H}$ and each $F_H \in \mathcal{F}_H$, we have $F^* \cap E(G[H]) \subseteq F_H$, i.e., forced edges are part of all subsolutions. In our algorithm, we make use of the fact that H_i has common edges with H_{i-1} and H_{i-2}, but not with H_{i-3}. Let $D[i, F_1, F_2, F_3]$ denote the costs of using subsolutions $F_1 \in \mathcal{F}_{H_i}, F_2 \in \mathcal{F}_{H_{i+1}}, F_3 \in \mathcal{F}_{H_{i+2}}$ for habitats H_i, H_{i+1}, H_{i+2}, respectively, under the assumption that the cost for the subsolutions for all habitats $H_1, \ldots, H_{i+1}$ is minimal given F_1 and F_2. Given the values of all entries $D[j, \cdot, \cdot, \cdot]$ for $j < i$, the entry $D[i, F_1, F_2, F_3]$ can be computed as follows.

$$D[i, F_1, F_2, F_3] = \begin{cases} c(F_1 \cup F_2 \cup F_3), \text{ if } i = 1, \\ \min_{F_0 \in \mathcal{F}_{H_{i-1}}} D[i-1, F_0, F_1, F_2] \\ \qquad +c(F_3 \setminus (F_1 \cup F_2)), \text{ otherwise.} \end{cases}$$

The optimum cost for the instance is then

$$\min_{F_1 \in \mathcal{F}_{H_{r-2}}, F_2 \in \mathcal{F}_{H_{r-1}}, F_3 \in \mathcal{F}_{H_r}} D[r-2, F_1, F_2, F_3]. \qquad \square$$

We can extend the algorithm of Lemma 20 to instances where $\partial\mathcal{H}$ induces a cycle: Pick an arbitrary habitat H and guess a solution $F_H \in \mathcal{F}_H$. Mark all edges in F_H as forced and run the algorithm on $\partial\mathcal{H} - H$, which induces a path. Theorem 3 then follows from Lemmas 11, 12, 19 and 20.

6 MAXIMUM DEGREE FIVE

We complement our results with (almost) tight hardness results via polynomial-time reductions from the NP-hard [19] Planar Cubic Vertex Cover problem, where, given a planar, connected graph $G = (V, E)$ where every vertex has degree three and an integer k, the task is to decide whether G has a vertex cover of size at most k.

Theorem 4 (★). 2-Diam GBP *is* NP*-hard even if (i) the graph has maximum degree five and the maximum habitat size is three, or (ii) the graph is planar, has maximum degree five, and the maximum habitat size is four.*

We sketch a reduction for the proof of Theorem 4(i) and defer the details to the paper's full version [40]. The construction uses vertex and edge gadgets as shown in Figure 6 (left and center). Every triangle forms a habitat. These are then connected by a so-called *docking* operation for a vertex gadget and an edge gadget (Figure 6 right): We *dock* the corresponding vertex and edge gadgets by identifying one of the dashed edges of each so that the directions of the arrows on the edges match (otherwise the degree bound is violated). The budget is chosen such that we can add four edges per vertex gadget and per edge gadget, plus an additional k edges.

To provide intuition for the correctness, if a vertex u is included in a vertex cover, then we choose $F_u^\top$ to reconnect all habitats in G_u, and $F_u^\perp$ otherwise. Although $F_u^\top$ contains one more edge than $F_u^\perp$, it covers all docking edges. So, for each edge gadget adjacent to G_u, we need one fewer edge to reconnect all habitats therein. As we only have budget for four edges per edge gadget, we require that each edge gadget must be docked to a vertex gadget that is in the vertex cover.

The direction constraint may violate planarity. Thus, for Theorem 4(ii), we require two different edge gadgets (see Figure 7) such that the docking pairs point in the same direction in one and in different directions in the other. This is only possible with size-four habitats. See the full version of the paper [40] for details.

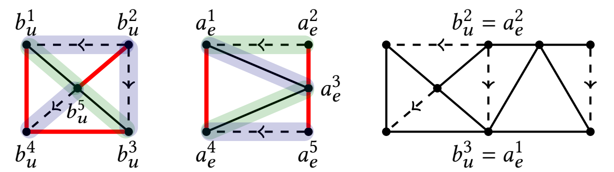

Figure 6: Illustrations for the construction for Theorem 4(i). *Left:* A vertex gadget for a vertex u. Red edges are size-two habitats (and thus forced). Blue edges mark the set $F_u^\top$. Green edges mark the set $F_u^\perp$. Dashed edges mark the docking pairs. *Center:* An edge gadget for an edge $e = \{u, v\}$. Green edges mark the set F_e^u. Blue edges mark the set F_e^v. *Right:* The result of docking the two gadgets at (b_u^2, b_u^3) and (a_e^2, a_e^1).

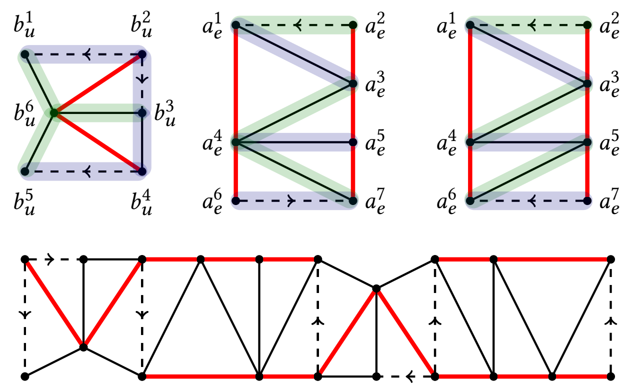

Figure 7: Illustrations of the construction for Theorem 4(ii). Red edges are forced. Green and blue highlights are two (sub)solutions. *Top Left:* A vertex gadget for a vertex u. *Top Center and Right:* A default edge gadget for an edge e. *Top Right:* An anti-crossing edge gadget. *Bottom:* The result of docking two vertex and two edge gadgets; the right edge gadget is anti-crossing.

7 EPILOGUE

We showed that reconnecting habitats so that every habitat has diameter two is computationally challenging even in sparse graphs with small maximum habitat sizes. On the way, we also identified intriguing structural properties to enable efficient algorithms.

The main unanswered question in this work is whether 2-Diam GBP-C is polynomial-time solvable on graphs with maximum degree four and maximum habitat size at least five. We believe that it is possible to extend our habitat intersection graph approach. If there are habitats of size six, we can, however, find another proper docking structure that can also result in paths. We also note that the dynamic programming approach for paths in the habitat intersection graph can be generalized to trees and to graphs of bounded treewidth. This raises the question: What do instances with these habitat intersection structures look like?

Finally, one limitation of our model is that every habitat must have diameter $d = 2$. Future work could extend our study by considering the cases $d \geq 3$, or even considering an adaptive diameter bound, for example, $\mathrm{diam}(G[F][H]) \leq \alpha\, \mathrm{diam}(G[H])$ for $\alpha \geq 1$.

ACKNOWLEDGMENTS

Till Fluschnik acknowledges support by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project PACS (FL 1247/1-1, 522475669).

REFERENCES

[1] Florian Adriaens and Aristides Gionis. 2022. Diameter Minimization by Shortcutting with Degree Constraints. In *Proc. of 22nd ICDM*. IEEE Computer Society, 843–848.

[2] Deepesh Agarwal, Julio-Cesar Silva Araujo, Christelle Caillouet, Frederic Cazals, David Coudert, and Stephane Pérennes. 2013. Connectivity Inference in Mass Spectrometry Based Structure Determination. In *Proc. of 21st ESA*. Springer, 289–300.

[3] Dana Angluin, James Aspnes, and Lev Reyzin. 2010. Inferring Social Networks from Outbreaks. In *Proc. of 21st ALT*. Springer, 104–118.

[4] Mohammadhossein Bateni, Mohammadtaghi Hajiaghayi, and Dániel Marx. 2011. Approximation Schemes for Steiner Forest on Planar Graphs and Graphs of Bounded Treewidth. *J. ACM* 58, 5 (2011), 1–37.

[5] Victoria J Bennett. 2017. Effects of road density and pattern on the conservation of species and biodiversity. *Current Landscape Ecology Reports* 2, 1 (2017), 1–11.

[6] Davide Bilò, Luciano Gualà, and Guido Proietti. 2012. Improved approximability and non-approximability results for graph diameter decreasing problems. *Theor. Comput. Sci.* 417 (2012), 12–22.

[7] Ulrik Brandes, Sabine Cornelsen, Barbara Pampel, and Arnaud Sallaberry. 2012. Path-Based Supports for Hypergraphs. *J. Discrete Algorithms* 14 (2012), 248–261.

[8] Kevin Buchin, Marc J. van Kreveld, Henk Meijer, Bettina Speckmann, and Kevin Verbeek. 2011. On Planar Supports for Hypergraphs. *J. Graph Algorithms Appl.* 15, 4 (2011), 533–549.

[9] Jiehua Chen, Christian Komusiewicz, Rolf Niedermeier, Manuel Sorge, Ondřej Suchý, and Mathias Weller. 2015. Polynomial-Time Data Reduction for the Subset Interconnection Design Problem. *SIAM J. Discrete Math.* 29, 1 (2015), 1–25.

[10] Gregory Chockler, Roie Melamed, Yoav Tock, and Roman Vitenberg. 2007. Constructing Scalable Overlays for Pub-Sub with Many Topics. In *Proc. of 26th PODC*. ACM, 109–118.

[11] Vincent Conitzer, Jonathan Derryberry, and Tuomas Sandholm. 2004. Combinatorial Auctions with Structured Item Graphs. In *Proc. of 19th AAAI*. AAAI Press, 212–218.

[12] Erik D. Demaine and Morteza Zadimoghaddam. 2010. Minimizing the Diameter of a Network Using Shortcut Edges. In *Proc. of 12th SWAT*. 420–431.

[13] Ding-Zhu Du and Zevi Miller. 1988. Matroids and Subset Interconnection Design. *SIAM J. Discrete Math.* 1, 4 (1988), 416–424.

[14] Ding-Zhu Du and Dean F. Kelley. 1995. On Complexity of Subset Interconnection Designs. *J. Glob. Optim.* 6 (1995), 193–205.

[15] Bernard Elspas. 1964. Topological Constraints on Interconnection-Limited Logic. In *Proc. of 5th SWCT*. 133–137.

[16] Hongbing Fan, Christian Hundt, Yu-Liang Wu, and Jason Ernst. 2008. Algorithms and Implementation for Interconnection Graph Problem. In *Proc. 2nd COCOA*. Springer, 201–210.

[17] Till Fluschnik and Leon Kellerhals. 2024. Placing Green Bridges Optimally, with a Multivariate Analysis. *Theory Comput. Syst.* 68, 5 (2024), 1312–1338.

[18] Fabrizio Frati, Serge Gaspers, Joachim Gudmundsson, and Luke Mathieson. 2015. Augmenting Graphs to Minimize the Diameter. *Algorithmica* 72 (2015), 995–1010.

[19] M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. 1976. Some Simplified NP-Complete Graph Problems. *Theor. Comput. Sci.* 1, 3 (1976), 237–267.

[20] Aristides Gionis, Polina Rozenshtein, Nikolaj Tatti, and Evimaria Terzi. 2017. Community-Aware Network Sparsification. In *Proc. of 17th SDM*. SIAM, 426–434.

[21] Andrew J. Hamer and Mark J. McDonnell. 2008. Amphibian ecology and conservation in the urbanising world: A review. *Biol. Conserv.* 141, 10 (2008), 2432–2449.

[22] Lori A Hennings and Jonathan Andrew Soll. 2010. *Wildlife corridors and permeability: a literature review.* Metro Sustainability Center.

[23] Maike Herkenrath, Till Fluschnik, Francesco Grothe, and Leon Kellerhals. 2022. Placing Green Bridges Optimally, with Habitats Inducing Cycles. In *Proc. of 31st IJCAI*. IJCAI Organization, 3825–3831.

[24] Emanuel Herrendorf, Christian Komusiewicz, Nils Morawietz, and Frank Sommer. 2024. On the Complexity of Community-Aware Network Sparsification. In *Proc. of 49th MFCS (LIPIcs, Vol. 306)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 60:1–60:18.

[25] Jun Hosoda, Juraj Hromkovič, Taisuke Izumi, Hirotaka Ono, Monika Steinová, and Koichi Wada. 2012. On the Approximability and Hardness of Minimum Topic Connected Overlay and Its Special Instances. *Theor. Comput. Sci.* 429 (2012), 144–154.

[26] Marcel P. Huijser, John W. Duffield, Anthony P. Clevenger, Robert J. Ament, and Pat T. McGowen. 2009. Cost–Benefit Analyses of Mitigation Measures Aimed at Reducing Collisions with Large Ungulates in the United States and Canada: a Decision Support Tool. *Ecol. Soc.* 14, 2 (2009).

[27] Marcel P. Huijser, Pat T. McGowen, Amanda Hardy, Angela Kociolek, Anthony P. Clevenger, Dan Smith, and Robert J. Ament. 2008. Wildlife-vehicle collision reduction study: Report to congress. (2008).

[28] Jesper Jansson, Christos Levcopoulos, and Andrzej Lingas. 2023. Online and Approximate Network Construction from Bounded Connectivity Constraints. *Int. J. Found. Comput. Sci.* 34, 5 (2023), 453–468.

[29] Boris Klemz, Tamara Mchedlidze, and Martin Nöllenburg. 2014. Minimum Tree Supports for Hypergraphs and Low-Concurrency Euler Diagrams. In *Proc. of 14th SWAT*. Springer, 265–276.

[30] Ephraim Korach and Michal Stern. 2003. The Clustering Matroid and the Optimal Clustering Tree. *Math. Program.* 98, 1 (2003), 385–414.

[31] Ephraim Korach and Michal Stern. 2008. The Complete Optimal Stars-Clustering-Tree Problem. *Discrete Appl. Math.* 156, 4 (2008), 444–450.

[32] Katherine J. Lai, Carla P. Gomes, Michael K. Schwartz, Kevin S. McKelvey, David E. Calkin, and Claire A. Montgomery. 2011. The Steiner Multigraph Problem: Wildlife Corridor Design for Multiple Species. In *Proc. of 25th AAAI*. AAAI Press, 1357–1364.

[33] Ronan LeBras, Bistra Dilkina, Yexiang Xue, Carla P. Gomes, Kevin S. McKelvey, Michael K. Schwartz, and Claire A. Montgomery. 2013. Robust Network Design for Multispecies Conservation. In *Proc. of 27th AAAI*. AAAI Press, 1305–1312.

[34] Melih Onus and Andréa W. Richa. 2011. Minimum Maximum-Degree Publish-Subscribe Overlay Network Design. *IEEE/ACM Trans. Netw.* 19, 5 (2011), 1331–1343.

[35] Melih Onus and Andréa W. Richa. 2016. Parameterized Maximum and Average Degree Approximation in Topic-Based Publish-Subscribe Overlay Network Design. *Comput. Netw.* 94 (2016), 307–317.

[36] Kasim Öztoprak, Sina Layazali, and Erdogan Dogdu. 2017. Topic Distribution Constant Diameter Overlay Design Algorithm (TD-CD-ODA). In *Proc. of 11th ICSC*. IEEE, 482–487.

[37] J. Plesnik. 1981. The Complexity of Designing a Network with Minimum Diameter. *Networks* 11, 1 (1981), 77–85.

[38] A. A. Schoone, H. L. Bodlaender, and J. Van Leeuwen. 1987. Diameter Increase Caused by Edge Deletion. *J. Graph Theory* 11, 3 (1987), 409–427.

[39] Fraser Shilling, David Waetjen, Graham Porter, Claire Short, Morgen Karcs, Tiana Honigman, Mireya Mejrano, Gracyna Mohabir, Min Jyaw, Anna Jones, Winston Vickers, and Kathryn Harrold. 2021. *From Wildlife-Vehicle Conflict to Solutions for California Drivers & Animals.* Technical Report. Road Ecology Center, UC Davis.

[40] Christian Wallisch, Till Fluschnik, and Leon Kellerhals. 2025. Placing Green Bridges Optimally, with Close-Range Habitats in Sparse Graphs. *CoRR* abs/2510.21540 (2025).