C_{2k+1}-Coloring of Bounded-Diameter Graphs

Marta Piecyk
Warsaw University of Technology, Poland

Abstract

For a fixed graph H, in the graph homomorphism problem, denoted by Hom(H), we are given a graph G and we have to determine whether there exists an edge-preserving mapping $\varphi : V(G) \to V(H)$. Note that Hom(C_3), where C_3 is the cycle of length 3, is equivalent to 3-Coloring. The question of whether 3-Coloring is polynomial-time solvable on diameter-2 graphs is a well-known open problem. In this paper we study the Hom(C_{2k+1}) problem on bounded-diameter graphs for $k \geq 2$, so we consider all other odd cycles than C_3. We prove that for $k \geq 2$, the Hom(C_{2k+1}) problem is polynomial-time solvable on diameter-$(k + 1)$ graphs – note that such a result for $k = 1$ would be precisely a polynomial-time algorithm for 3-Coloring of diameter-2 graphs. Furthermore, we give subexponential-time algorithms for diameter-$(k + 2)$ and -$(k + 3)$ graphs.

We complement these results with a lower bound for diameter-$(2k + 2)$ graphs – in this class of graphs the Hom(C_{2k+1}) problem is NP-hard and cannot be solved in subexponential-time, unless the ETH fails.

Finally, we consider another direction of generalizing 3-Coloring on diameter-2 graphs. We consider other target graphs H than odd cycles but we restrict ourselves to diameter 2. We show that if H is triangle-free, then Hom(H) is polynomial-time solvable on diameter-2 graphs.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Mathematics of computing → Graph coloring

Keywords and phrases graph homomorphism, odd cycles, diameter

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.78

Related Version *Full Version*: https://doi.org/10.48550/arXiv.2403.06694 [29]

Funding *Marta Piecyk*: This research was funded by Polish National Science Centre, grant no. 2022/45/N/ST6/00237.

Acknowledgements We are grateful to Paweł Rzążewski for inspiring and fruitful discussions.

1 Introduction

A natural approach to computationally hard problems is to restrict the class of input graphs, for example by bounding some parameters. One of such parameters is the diameter, i.e., for a graph G, its diameter is the least integer d such that for every pair of vertices u, v of G, there is a u-v path on at most d edges. We say that G is a diameter-d graph, if its diameter is at most d. Recently, bounded-diameter graphs received a lot of attention [23, 20, 25, 3, 2, 24, 9, 6]. It is known that graphs from real life applications often have bounded diameter, for instance social networks tend to have bounded diameter [30]. Furthermore, almost all graphs have diameter 2, i.e., the probability that a random graph on n vertices has diameter 2 tends to 1 when n tends to infinity [21]. Therefore, solving a problem on bounded-diameter graphs captures a wide class of graphs. On the other hand, not all of the standard approaches can be used – note that the class of diameter-d graphs is not closed under vertex deletion.

Even if we consider the class of diameter-2 graphs, its members can contain any graph as an induced subgraph. Indeed, consider any graph G, and let G^+ be the graph obtained from G by adding a universal vertex u, i.e., we add vertex u and make it adjacent to all vertices of G. It is straightforward to observe that the diameter of G^+ is at most 2. This construction

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 78; pp. 78:1–78:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

can be used for many graph problems on diameter-2 graphs as a hardness reduction, which proves that they cannot be solved in subexponential time under the Exponential Time Hypothesis (ETH, see [19]), for instance Max Independent Set.

The construction of G^+ also gives us a reduction from $(k-1)$-Coloring to k-Coloring on diameter-2 graphs, and thus for any $k \geq 4$, the k-Coloring problem on diameter-2 graphs is NP-hard and cannot be solved in subexponential time, unless the ETH fails. Note that this argument does not work for $k = 3$ since then we reduce from 2-Coloring, which is polynomial-time solvable. A textbook reduction from NAE-SAT implies that for $d \geq 4$, the 3-Coloring problem is NP-hard and cannot be solved in subexponential-time, unless the ETH fails [28]. Therefore, it is only interesting to study 3-Coloring on diameter-2 and -3 graphs. Mertzios and Spirakis proved that 3-Coloring is NP-hard on diameter-3 graphs [25]. However, the question of whether 3-Coloring can be solved in polynomial time on diameter-2 graphs remains open.

For 3-Coloring on diameter-2 graphs, subexponential-time algorithms were given, first by Mertzios and Spirakis with running time $2^{\mathcal{O}(\sqrt{n \log n})}$ [25]. This was later improved by Dębski, Piecyk, and Rzążewski, who gave an algorithm with running time $2^{\mathcal{O}(n^{1/3} \cdot \log^2 n)}$ [9]. They also provided a subexponential-time algorithm for 3-Coloring for diameter-3 graphs.

The 3-Coloring problem on bounded-diameter graphs was also intensively studied on instances with some additional restrictions, i.e., on graphs with some forbidden induced subgraphs – and on such graph classes polynomial-time algorithms were given [23, 20, 24].

One of the generalizations of graph coloring are homomorphisms. For a fixed graph H, in the graph homomorphism problem, denoted by Hom(H), we are given a graph G, and we have to determine whether there exists an edge-preserving mapping $\varphi : V(G) \to V(H)$, i.e., for every $uv \in E(G)$, it holds that $\varphi(u)\varphi(v) \in E(H)$. Observe that for K_k being a complete graph on k vertices, the Hom(K_k) problem is equivalent to k-Coloring. Observe also that the problem is trivial when H contains a vertex x with a loop since we can map all vertices of G to x. In case when H is bipartite, in fact we have to verify whether G is bipartite and this can be done in polynomial time. Hell and Nešetřil proved that for all other graphs H, i.e., loopless and non-bipartite, the Hom(H) problem is NP-hard [18]. Such a complete dichotomy was provided by Feder, Hell, and Huang also for the list version of the problem [13]. The graph homomorphism problem and its variants in various graph classes and under various parametrizations received recently a lot of attention [26, 15, 14, 4, 5, 7, 8, 14, 27, 17]. We also point out that among all target graphs H, odd cycles received a lot of attention [16, 1, 10, 31, 22]. Note that the cycle on 5 vertices is the smallest graph H such that the Hom(H) problem is not equivalent to graph coloring.

Our contribution. In this paper we consider the Hom(C_{2k+1}) problem on bounded-diameter graphs, where C_{2k+1} denotes the cycle on $2k+1$ vertices. Note that for $k = 1$, we have C_3, so this problem is equivalent to 3-Coloring. In this work we consider all other values of k. Our first result is the following.

▶ **Theorem 1.** *Let $k \geq 2$. Then Hom(C_{2k+1}) can be solved in polynomial time on diameter-$(k+1)$ graphs.*

Note that such a result for $k = 1$ would yield a polynomial-time algorithm for 3-Coloring on diameter-2 graphs. Let us discuss the crucial points where this algorithm cannot be applied directly for $k = 1$. The first property, which holds for every cycle except C_3 and C_6, is that if for some set S of vertices, any two of them have a common neighbor, then there is a vertex that is a common neighbor of all vertices of S. Furthermore, for every cycle of length at least 5 except C_6, for a set S of 3 vertices, every vertex of S has a private neighbor with respect to S, i.e., a neighbor that is non-adjacent to any other vertex of S.

Table 1 Complexity of $\textsc{Hom}(C_{2k+1})$ on bounded-diameter graphs. The symbol in the cell (k, d) denotes that $\textsc{Hom}(C_{2k+1})$ on diameter-d, resp., P – is polynomial-time solvable, NP-h – is NP-hard, S – can be solved in subexponential time, and NS – cannot be solved in subexponential time under the ETH. The rows for $k \geq 2$ are filled due to Theorems 1–3. The first row is based on [25, 9].

k / diam	1	2	3	4	5	6	7	8	9	≥ 10
1	P	S	S, NP-h	NS	NS	NS	NS	NS	NS	NS
2	P	P	P	S	S	NS	NS	NS	NS	NS
3	P	P	P	P	S	S	?	?	NS	NS
4	P	P	P	P	P	S	S	?	?	NS
≥ 5	P	P	P	P	P	P	S	S	?	?

We first show that for an instance of $\textsc{Hom}(C_{2k+1})$, for each vertex v we can deduce the set of vertices it can be mapped to and define a list of v – all lists are of size at most 3. The properties discussed above allow us to encode coloring of a vertex with list of size 3 using its neighbors with lists of size two, and such a reduced instance of a slightly more general problem (we have more constraints than just the edges, but all of them are binary) can be solved in polynomial time by reduction to 2-SAT, similar to the one of Edwards [11].

Furthermore, we give subexponential-time algorithms.

▶ **Theorem 2.** *Let $k \geq 2$. Then $\textsc{Hom}(C_{2k+1})$ can be solved in time:*

(1.) $2^{\mathcal{O}((n \log n)^{\frac{k+1}{k+2}})}$ *on diameter-$(k+2)$ n-vertex graphs,*

(2.) $2^{\mathcal{O}((n \log n)^{\frac{k+2}{k+3}})}$ *on diameter-$(k+3)$ n-vertex graphs.*

Here the branching part of the algorithm is rather standard. The more involved part is to show that after applying braching and reduction rules we are left with an instance that can be solved in polynomial time. Similar to Theorem 1, we first analyze the lists of all vertices, and then reduce to an instance of more general problem where all lists are of size at most 2.

We complement Theorem 1 and Theorem 2 with the following NP-hardness result – since our reduction from 3-SAT is linear, we also prove that the problem cannot be solved in subexponential time under the ETH. The summary of the results is presented in Table 1.

▶ **Theorem 3 (♠).** *Let $k \geq 2$. Then $\textsc{Hom}(C_{2k+1})$ is NP-hard on diameter-$(2k+2)$ graphs (of radius $k+1$) and cannot be solved in subexponential time, i.e., there is no algorithm solving every n-vertex instance G of $\textsc{Hom}(C_{2k+1})$ in time $2^{o(n)} \cdot n^{\mathcal{O}(1)}$, unless the ETH fails.*

The next direction we study in the paper is the following. Instead of considering larger odd cycles and apropriate diameter, we focus on diameter-2 input graphs, but we change the target graph to arbitrary H. Note that it only makes sense to consider graphs H of diameter-2, since the homomorphic image of a diameter-2 graph has to induce a diameter-2 subgraph. We consider triangle-free target graphs. We point out that the class of triangle-free diameter-2 graphs is still very rich, for instance, contains all Mycielski graphs.

▶ **Theorem 4.** *Let H be a triangle-free graph. Then the $\textsc{Hom}(H)$ problem can be solved in polynomial time on diameter-2 graphs.*

Finally, let us point out that we actually prove stronger statements of Theorem 1, Theorem 2, Theorem 4 as we consider more general list version of the problem.

The proofs of statements marked with ♠ can be found in the full version of the paper [29].

2 Preliminaries

For a vertex v, by $N_G(v)$ we denote the neighborhood of v in G, and for a set $U \subseteq V(G)$, we denote $N_G(U) := \bigcup_{u \in U} N_G(u) \setminus U$. By $\text{dist}_G(u,v)$ we denote the length (number of edges) of a shortest u-v path in G. For a positive integer d, by $N_G^{\leq d}(v)$ we denote the set of all vertices $u \in V(G)$ such that $\text{dist}_G(u,v) \leq d$. If G is clear from the contex, we omit the subscript G and simply write $N(v)$, $N(U)$, $N^{\leq d}(v)$, and $\text{dist}(u,v)$. A *diameter* of G, denoted by $\text{diam}(G)$, is the maximum $\text{dist}(u,v)$ over all pairs of vertices $u,v \in V(G)$. We say that G is *diameter-d* graph if $\text{diam}(G) \leq d$. A *radius* of G is the minimum integer r such that there is a vertex $z \in V(G)$ such that for every $v \in V(G)$, it holds that $\text{dist}(v,z) \leq r$. By $[n]$ we denote the set $\{1,2,\ldots,n\}$ and by $[n]_0$ we denote $\{0,1,\ldots,n\}$. Throughout this paper all graphs we consider are simple, i.e., no loops, no multiple edges.

Homomorphisms. For graphs G, H, a homomorphism from G to H is an edge-preserving mapping $\varphi : V(G) \to V(H)$, i.e., for every $uv \in E(G)$, it holds $\varphi(u)\varphi(v) \in E(H)$. For fixed H, called *target*, in the homomorphism problem, denoted by $\text{Hom}(H)$, we are given a graph G, and we have to determine whether there exists a homomorphism from G to H. In the list homomorphism problem, denoted by $\text{LHom}(H)$, G is given along with lists $L : V(G) \to 2^{V(H)}$, and we have to determine if there is a homomorphism φ from G to H which additionally respects lists, i.e., for every $v \in V(G)$ it holds $\varphi(v) \in L(v)$. We will write $\varphi : G \to H$ (resp. $\varphi : (G,L) \to H$) if φ is a (list) homomorphism from G to H, and $G \to H$ (resp. $(G,L) \to H$) to indicate that such a (list) homomorphism exists. Since the graph homomorphism problem generalizes graph coloring we will often refer to homomorphism as coloring and to vertices of H as colors. For an instance (G,L) and an induced subgraph G' of G while refering to a subinstance $(G', L|_{G'})$, we will often simply write (G', L).

Cycles. Whenever C_{2k+1} is the target graph, we will denote its vertex set by $[2k]_0$, unless stated explicitly otherwise. Moreover, whenever we refer to the vertices of the $(2k+1)$-cycle, i.e., cycle on $2k+1$ vertices, by $+$ and $-$ we denote respectively the addition and the subtraction modulo $2k+1$.

Lists. For an instance (G,L) of $\text{LHom}(C_{2k+1})$, by V_i we denote the set of vertices v of G such that $|L(v)| = i$. Sometimes we will refer to vertices of V_1 as *precolored vertices*. We also define $V_{\geq i} = \bigcup_{j \geq i} V_j$. We say that a list $L(v)$ is *of type* $(\ell_1, \ldots, \ell_r)$ if $|L(v)| = r+1$ and its vertices can be ordered $c_0, \ldots, c_r$ so that for every $i \in [r-1]_0$, we have that $c_{i+1} = c_i + \ell_i$. For example, for $k \geq 4$, one of the types of the list $\{1,4,6,7\}$ is $(3,2,1)$.

Binary CSP and 2-SAT. For a given set (domain) D, in the Binary Constraint Satisfaction problem (BCSP) we are given a set V of variables, list function $L : V \to 2^D$, and constraint function $C : V \times V \to 2^{D \times D}$. The task is to determine whether there exists an assignment $f : V \to D$ such that for every $v \in V$, we have $f(v) \in L(v)$ and for every pair $(u,v) \in V \times V$, we have $(f(u), f(v)) \in C(u,v)$. Clearly, any instance of $\text{LHom}(H)$ can be seen as an instance of BCSP, where the domain D is $V(H)$, list function remains the same, and for every edge $uv \in E(G)$ we set $C(u,v) = \{(x,y) \mid xy \in E(H)\}$ and for every $uv \notin E(G)$ we set $C(u,v) = V(H) \times V(H)$. We will denote by $\text{BCSP}(H,G,L)$ the instance of BCSP corresponding to the instance (G,L) of $\text{LHom}(H)$. Standard approach of Edwards [11] with a reduction to 2-SAT implies that in polynomial time we can solve an instance of BCSP with all list of size at most two.

▶ **Theorem 5** (Edwards [11]). *Let (V, L, C) be an instance of BCSP over the domain D. Assume that for every $v \in V$ it holds $|L(v)| \leq 2$. Then we can solve the instance (V, L, C) in polynomial time.*

3 Reduction rules and basic observations

In this section we define reduction rules and show some basic observations.

Reduction rules

Let H be a graph and let (G, L) be an instance of LHom(H). We define the following reduction rules.

(R1) If $H = C_{2k+1}$ and G contains an odd cycle of length at most $2k-1$, then return NO.

(R2) If $H = C_{2k+1}$ and in G there are two $(2k+1)$-cycles with consecutive vertices respectively $c_0, \ldots, c_{2k}$ and $c'_0, \ldots, c'_{2k}$ and such that $c_0 = c'_0$ and $c_i = c'_j$ for some $i, j \neq 0$, then a) if $i = j$, then identify c_ℓ with c'_ℓ for every $\ell \in [2k]$, b) if $i = -j$, then identify c_ℓ with $c'_{(-\ell)}$ for every $\ell \in [2k]$, c) otherwise return NO.

(R3) For every edge uv, if there is $x \in L(u)$ such that $N_H(x) \cap L(v) = \emptyset$, then remove x from $L(u)$.

(R4) If there is $v \in V(G)$ such that $L(v) = \emptyset$, then return NO.

(R5) For every $v \in V(G)$, if there are $x, y \in L(v)$ such that for every $u \in N_G(v)$, we have $N_H(x) \cap L(u) \subseteq N_H(y) \cap L(u)$, then remove x from $L(v)$.

(R6) For a vertex $x \in V(H)$, and vertices $u, v \in V(G)$ such that $L(u) = L(v) = \{x\}$, if $uv \in E(G)$, then return NO, otherwise contract u with v.

Clearly, each of the above reduction rules can be applied in polynomial time. The following lemma shows that the reduction rules are safe.

▶ **Lemma 6.** *After applying each reduction rule to an instance (G, L) of LHom(H), we obtain an equivalent instance with diameter at most* $\mathrm{diam}(G)$.

Proof. First, any odd cycle cannot be mapped to a larger odd cycle, so the reduction rule (R1) is safe. Furthermore, for any $(2k+1)$-cycle, in any list homomorphism to C_{2k+1}, its consecutive vertices have to be mapped to consecutive vertices of C_{2k+1}. Let $c_0, \ldots, c_{2k}$ and $c'_0, \ldots, c'_{2k}$ be the vertices of two $(2k+1)$-cycles such that $c_0 = c'_0$ and $c_i = c'_j$ for some $i, j \neq 0$. Suppose we are dealing with a yes-instance and let $\varphi : (G, L) \to C_{2k+1}$. Without loss of generality assume that $\varphi(c_0) = \varphi(c'_0) = 0$ and $\varphi(c_i) = \varphi(c'_j) = i$. Then $\varphi(c_s) = s$ for every $s \in [2k]_0$. Moreover, either $j = i$ or $j = -i$, and $\varphi(c'_s) = s$ for every $s \in [2k]_0$ in the first case, or $\varphi(c'_s) = -s$ for every $s \in [2k]_0$ in the second case. Therefore, the reduction rule (R2) is safe.

Let $uv \in E(G)$ be such that there is $x \in L(u)$ such that $N(x) \cap L(v) = \emptyset$. Suppose that there is a list homomorphism $\varphi : (G, L) \to C_{2k+1}$ such that $\varphi(u) = x$. Then v must be mapped to a vertex from $N(x) \cap L(v) = \emptyset$, a contradiction. Thus we can safely remove x from $L(u)$, and (R3) is safe. Clearly, if any list of a vertex is an empty set, then we are dealing with a no-instance and thus (R4) is safe. Finally, assume there is $v \in V(G)$ and $x, y \in L(v)$ such that for every $u \in N_G(v)$, it holds $N_H(x) \cap L(u) \subseteq N_H(y) \cap L(u)$, and suppose there is a list homomorphism $\varphi : (G, L) \to H$ such that $\varphi(v) = x$. Then φ' defined so that $\varphi'(v) = y$ and $\varphi'(w) = \varphi(w)$ for $w \in V(G) \setminus \{v\}$ is also a list homomorphism $(G, L) \to H$. Therefore, (R5) is safe. If two vertices have the same one-element list, then they must be mapped to the same vertex. Since we only consider loopless graphs H, adjacent vertices of G cannot be mapped to the same vertex. Therefore, if two vertices u, v of G have lists $L(v) = L(u) = \{x\}$ for some $x \in V(H)$, then if $uv \in E(G)$ we are dealing with a no-instance. Otherwise, we can identify u and v and thus (R6) is safe. ◀

In the following lemma we describe the lists of vertices that are at some small distance of a precolored vertex.

▶ **Lemma 7.** *Let $k \geq 2$ and let (G, L) be an instance of* LHOM(C_{2k+1})*. Let $u \in V(G)$ be such that $L(u) = \{i\}$ and let $v \in V(G)$ be such that $\mathrm{dist}(u, v) = d$. If none of the reduction rules can be applied, then*

a) $L(v) \subseteq \{i - d, i - d + 2, \ldots, i - 2, i, i + 2, \ldots, i + d - 2, i + d\}$ *if d is even,*

b) $L(v) \subseteq \{i - d, i - d + 2, \ldots, i - 1, i + 1, \ldots, i + d - 2, i + d\}$ *if d is odd.*

Proof. Let P be a shortest u-v path such that the consecutive vertices of P are $u = p_0, p_1, \ldots, p_d = v$. We have $L(p_0) = \{i\}$. Since the reduction rule (R3) cannot be applied for p_0p_1, we must have $L(p_1) \subseteq \{i - 1, i + 1\}$. Applying this reasoning to consecutive vertices of the path, for $j \in [d]$, we must have

$$L(v) \subseteq \{i - j, i - j + 2, \ldots, i - 2, i, i + 2, \ldots, i + j - 2, i + j\},$$

if j is even,

$$L(v) \subseteq \{i - j, i - j + 2, \ldots, i - 1, i + 1, \ldots, i + j - 2, i + j\},$$

if j is odd, which completes the proof. ◀

The next lemma immediately follows from Lemma 7.

▶ **Lemma 8.** *Let $k \geq 2$ and let (G, L) be an instance of* LHOM(C_{2k+1})*. Let $u, w \in V(G)$ be such that $L(u) = \{i\}$ and $L(w) = \{i+1\}$. Let $v \in V(G)$ be such that $\mathrm{dist}(u, v) = \mathrm{dist}(w, v) = k + \ell$. If none of the reduction rules can be applied, then $L(v) \subseteq \{i + k - \ell + 1, i + k - \ell + 2, \ldots, i + k + \ell + 1\}$.*

In the following lemma we show that for a partial mapping $\varphi : V(G) \to [2k]_0$ for $k \geq 2$, for $v \in V(G)$, if every pair (a, b) of its neighbors is precolored so that $\varphi(a)$ and $\varphi(b)$ have a common neighbor in $L(v)$, then φ can be extended to v so that it preserves the edges containing v.

▶ **Lemma 9.** *Let $k \geq 2$, let (G, L) be an instance of* LHOM(C_{2k+1}) *and let $v \in V(G)$. Let $\varphi : N(v) \to [2k]_0$ be a mapping such that for every $a, b \in N(v)$, we have that $N_{C_{2k+1}}(\varphi(a)) \cap N_{C_{2k+1}}(\varphi(b)) \cap L(v) \neq \emptyset$. Then $\bigcap_{u \in N(v)} N_{C_{2k+1}}(\varphi(u)) \cap L(v) \neq \emptyset$.*

Proof. Define $A = \{\varphi(u) \mid u \in N(v)\}$. If $|A| \leq 2$, then the statement clearly follows. We will show that this is the only case. So suppose that $|A| \geq 3$. If two distinct vertices of C_{2k+1} for $k \geq 2$ have a common neighbor, then they must be at distance exactly two. Without loss of generality, let $0, 2 \in A$ and $1 \in L(v)$. Moreover, let $i \in A \setminus \{0, 2\}$. By assumption $N_{C_{2k+1}}(i) \cap N_{C_{2k+1}}(0) \neq \emptyset$, so $i = 2k - 1$. On the other hand, $N_{C_{2k+1}}(i) \cap N_{C_{2k+1}}(2) \neq \emptyset$, so $i = 4$. Thus $2k - 1 = 4$, a contradiction. ◀

In the next lemma we show that for an odd cycle C and a vertex v there is at least one pair of consecutive vertices of C with equal distances to v.

▶ **Lemma 10.** *Let G be a connected graph, let C be a cycle in G with consecutive vertices $c_0, \ldots, c_{2k}$, and let $v \in V(G) \setminus V(C)$. Then there is $i \in [2k]_0$ such that $\mathrm{dist}(v, c_i) = \mathrm{dist}(v, c_{i+1})$.*

Proof. First observe, that for all i, we have $|\operatorname{dist}(v, c_i) - \operatorname{dist}(v, c_{i+1})| \leq 1$ since $c_i c_{i+1} \in E(G)$. Therefore, going around the cycle the distance from v to c_i can increase by 1, decrease by 1, or remain the same. Since we have to end up with the same value at the end and the length of the cycle is odd, there is at least one pair of consecutive vertices c_i, c_{i+1} such that $\operatorname{dist}(v, c_i) = \operatorname{dist}(v, c_{i+1})$. ◀

4 Polynomial-time algorithm

In this section we prove Theorem 1. In fact we prove a stronger statement for the list version of the problem.

▶ **Theorem 11.** *Let $k \geq 2$. Then* LHom(C_{2k+1}) *can be solved in polynomial time on diameter-$(k+1)$ graphs.*

Proof. Let (G, L) be an instance of LHom(C_{2k+1}) such that G has diameter at most $k+1$. First for every $i \in [2k]_0$ we check whether there is a list homomorphism $\varphi : (G, L) \to C_{2k+1}$ such that no vertex is mapped to i, so we look for a list homomorphism to a path which can be done in polynomial time by [12].

If there is no such a list homomorphism, then we know that all colors have to be used and thus we guess $2k+1$ vertices that will be mapped to distinct vertices of C_{2k+1}. Let $c_0, \ldots, c_{2k}$ be the vertices such that c_i is precolored with i. We check whether such a partial assignment satisfies the edges with both endpoints precolored. Moreover, if c_i, c_{i+1} are non-adjacent, we add the edge $c_i c_{i+1}$ – this operation is safe as c_i, c_{i+1} are precolored with consecutive vertices of C_{2k+1} and adding an edge does not increase the diameter. Finally, we exhaustively apply the reduction rules.

Observe that the vertices $c_0, \ldots, c_{2k}$ induce a $(2k+1)$-cycle C. Suppose there is a vertex v that after the above procedure is not on C. By Lemma 10, there is $i \in [2k]_0$ such that $\operatorname{dist}(v, c_i) = \operatorname{dist}(v, c_{i+1}) =: \ell$.

First we show that we cannot have $\ell \leq k$. Suppose otherwise. Let P_1, P_2 be shortest v-c_i-, and v-c_{i+1}-paths, respectively. Let u be the their last common vertex (it cannot be c_i or c_{i+1} as the distances are the same and c_i, c_{i+1} are adjacent). Note that since P_1, P_2 are shortest, the u-c_i-path P_1' obtained from P_1 and the u-c_{i+1}-path P_2' obtained from P_2 have the same length. Therefore we can construct a cycle by taking P_1', P_2' and the edge $c_i c_{i+1}$. The length of the cycle is odd, and it is at most $2k+1$. This cycle contains at least two vertices from C, so either it should be contracted by (R2) or (R1) would return NO. Therefore we cannot have $\ell \leq k$, and thus, since $\operatorname{diam}(G) \leq k+1$, we have $\ell = k+1$.

By Lemma 8, for $v \in V_{\geq 3}$, we have that $L(v) \subseteq \{i+k, i+k+1, i+k+2\}$. Therefore, all lists of our instance have size at most 3. Moreover, each vertex of V_3 has list of type $(1, 1)$. Furthermore, since (R3) cannot be applied, for a vertex with list $\{j, j+1, j+2\}$, the possible lists of its neighbors in $G[V_3]$ are then $\{j-1, j, j+1\}$, $\{j, j+1, j+2\}$, and $\{j+1, j+2, j+3\}$.

For a list $\{i, j, r\}$ of type $(1, 1)$, where j is the vertex such that $j = i+1$ and $j = r-1$, we will call j the *middle vertex* of $\{i, j, r\}$. For a homomorphism φ we will say that a vertex $v \in V_3$ is *φ-middle*, if φ maps v to the middle vertex of its list.

Now consider a connected component S of $G[V_3]$, let $v \in V(S)$, and let $L(v) = \{j-1, j, j+1\}$. The following claim is straightforward.

▷ Claim 12. Suppose there is a list homomorphism $\varphi : (S, L) \to C_{2k+1}$. Then

(1.) if v is φ-middle, then any $u \in N_S(v)$ with list $\{j-1, j, j+1\}$ cannot be φ-middle, and every $w \in N_S(v)$ with list $\{j-2, j-1, j\}$ or $\{j, j+1, j+2\}$ has to be φ-middle,

(2.) if v is not φ-middle, then every $u \in N_S(v)$ with list $\{j-1, j, j+1\}$ has to be φ-middle, and any $w \in N_S(v)$ with list $\{j-2, j-1, j\}$ or $\{j, j+1, j+2\}$ cannot be φ-middle.

Thus deciding if one vertex of S is φ-middle, already determines for every vertex of S if it is φ-middle or not. It is described more formally in the following claim, whose proof can be found in the full version of the paper [29].

▷ Claim 13 (♠). In polynomial time we can either (1) construct a partition (U_1, U_2) of $V(S)$ (U_1, U_2 might be empty) such that for every list homomorphism $\varphi : (S, L) \to C_{2k+1}$, either all vertices of U_1 are φ-middle and no vertex of U_2 is φ-middle, or all vertices of U_2 are φ-middle and no vertex of U_1 is φ-middle, or (2) conclude that we are dealing with a no-instance.

Therefore, for every connected component S of $G[V_3]$ we solve two subinstances:

(I1) (S, L_1), where for every $v \in V(S)$ with list $\{i-1, i, i+1\}$ for some $i \in [2k]_0$, $L_1(v) = \{i\}$ if $v \in U_1$ and $L_1(v) = \{i-1, i+1\}$ if $v \in U_2$,

(I2) (S, L_2), where for every $v \in V(S)$ with list $\{i-1, i, i+1\}$ for some $i \in [2k]_0$, $L_2(v) = \{i\}$ if $v \in U_2$ and $L_2(v) = \{i-1, i+1\}$ if $v \in U_1$.

Note that both subinstances have all lists of size at most two and thus can be solved in polynomial time by Theorem 5. If for some component in both cases we obtain NO, then we return NO.

Creating a BCSP instance. Let $(V(G), L, C) = \text{BCSP}(C_{2k+1}, G[V_1 \cup V_2], L)$. We will modify now the instance $(V(G), L, C)$ so it is equivalent to the instance (G, L). For every $v \in V_3$ and for every pair of $a, b \in N(v) \cap V_2$, we leave in $C(a, b)$ only these pairs of vertices that have a common neighbor in $L(v)$ – recall that by Lemma 9 this ensures us that there will be color left for v. Furthermore, for every connected component S of $G[V_3]$, we add constraints according to which of the two possibilities S can be properly colored (possibly S can be colored in both cases) as follows. Let $v \in V(S)$ with $L(v) = \{i-1, i, i+1\}$, and without loss of generality assume that $v \in U_1$. The neighbors of v in V_2 have lists $\{i-1, i\}$ and $\{i, i+1\}$. For each such v:

- if S cannot be properly colored so that vertices of U_1 are middle (v is colored with i), we remove $i-1$ and $i+1$ from the lists of neigbors of v,
- if S cannot be properly colored so that vertices of U_1 are not middle (v is colored with one of $i-1, i+1$), we remove i from the lists of neigbors of v.
- Moreover, for every $u \in V(S)$ with list $\{j-1, j, j+1\}$, for every neighbor $u' \in V_2 \cap N(u)$, and for every $v' \in V_2 \cap N(v)$, if $u \in U_1$, then we remove from $C(u', v')$ pairs $(j, i+1)$, $(j, i-1)$, $(j-1, i)$, $(j+1, i)$, and if $u \in U_2$, then we remove from $C(u', v')$ the pairs (j, i), $(j-1, i-1)$, $(j-1, i+1)$, $(j+1, i-1)$, $(j+1, i+1)$.

This completes the construction of BCSP instance $(V(G), L, C)$. By Theorem 5 we solve $(V(G), L, C)$ in polynomial time.

Correctness. The detailed proof of correctness can be found in the full version of the paper [29].

▷ Claim 14 (♠). $(V(G), L, C)$ is a yes-instance of BCSP iff (G, L) is a yes-instance of LHom(C_{2k+1}).

Let us only mention here the idea behind each of introduced constraints. First, we start with $\text{BCSP}(C_{2k+1}, G[V_1 \cup V_2], L)$, so that the assignment restricted to $V_1 \cup V_2$ can be a list homomorphism φ on $(G[V_1 \cup V_2], L)$. The remaining constraints ensure us that we can extend φ to each connected component S of $G[V_3]$. First, we make sure that for every vertex $v \in V(S)$, there is a color left for v in $L(v)$ – it is a necessary condition. Observe that for a vertex v with list $\{i-1, i, i+1\}$, if the neighbors of v in V_2 are mapped so that there is a color left for v, then either they are colored with $i-1, i+1$ and the color left for v is i, or all such neighbors are colored with i, and both $i-1, i+1$ are left for v (unless one of them is not on $L(v)$). Therefore, we can assume that v has the same list as in one of the instances (I1) and (I2). Moreover, because of the last introduced constraint, all vertices of S have lists corresponding to the same instance, say (I1). Finally, if these lists are still present, then we know that (I1) is a yes-instance, and φ can be extended to S. This completes the proof. ◀

5 Subexponential-time algorithms

In this section we prove the following stronger version of Theorem 2.

▶ **Theorem 15.** *Let $k \geq 3$. Then* LHOM(C_{2k+1}) *can be solved in time:*

(1.) $2^{\mathcal{O}((n \log n)^{\frac{k+1}{k+2}})}$ *on n-vertex diameter-$(k+2)$ graphs,*

(2.) $2^{\mathcal{O}((n \log n)^{\frac{k+2}{k+3}})}$ *on n-vertex diameter-$(k+3)$ graphs.*

We start with defining branching rules crucial for our algorithm.

Branching rules. Let $k \geq 2$, let (G, L) be an instance of LHOM(C_{2k+1}) and let $d \geq \text{diam}(G)$. Let $\mu = \sum_{\ell=2}^{2k+1} \ell \cdot |V_\ell|$. We define the following branching rules.

(B1) If there is a vertex $v \in V_{\geq 2}$ with at least $(\mu \log \mu)^{1/d}$ neighbors in $V_{\geq 2}$, for some $a \in L(v)$, we branch on coloring v with a or not, i.e., we create two instances $I_a = (G, L_a)$, $I'_a = (G, L'_a)$ such that $L_a(u) = L'_a(u) = L(u)$ for every $u \in V(G) \backslash \{v\}$, and $L_a(v) = \{a\}$ and $L'_a(v) = L(v) \setminus \{a\}$.

(B2) We pick a vertex v and branch on the coloring of $N^{\leq d-1}[v] \cap V_{\geq 2}$, i.e., for every mapping f of $N^{\leq d-1}[v] \cap V_{\geq 2}$ that respects the lists, we create a new instance $I_f = (G, L_f)$ such that $L_f(u) = L(u)$ for $u \notin N^{\leq d-1}[v] \cap V_{\geq 2}$ and $L_f(w) = \{f(w)\}$ for $w \in N^{\leq d-1}[v] \cap V_{\geq 2}$.

Algorithm `Recursion Tree`

Let us describe an algorithm that for fixed d takes an instance (G, L) of LHOM(C_{2k+1}) with a fixed precolored $(2k+1)$-cycle C and such that $\text{diam}(G) \leq d$, and returns a rooted tree $\mathcal{R}$ whose nodes are labelled with subinstances of (G, L). We first introduce the root r of $\mathcal{R}$ and we label it with (G, L). Then for every node we proceed recursively as follows. Let s be a node labelled with an instance (G', L') of LHOM(C_{2k+1}). We first exhaustively apply to (G', L') reduction rules and if some of the reduction rules returns NO, then s does not have any children. Otherwise, if possible, we apply branching rule (B1). We choose $a \in L(v)$ for (B1) as follows. If on $N_{G'[V_{\geq 2}]}(v)$ there are no lists of type (2), then we take any $a \in L(v)$. Otherwise, let S be the most frequent list of type (2) on $N_{G'[V_{\geq 2}]}(v)$, and let $S = \{j-1, j+1\}$ for some $j \in [2k]_0$. Then we take any $a \in L(v) \setminus \{j\}$. After application of (B1), we exhaustively apply reduction rules to each instance. Furthermore, for each instance created by (B1), we create a child node of s and we label it with that instance.

So suppose that (B1) cannot be applied. We proceed as follows.

Case 1: there is no vertex v such that $\text{dist}(v, c_i) = \text{dist}(v, c_{i+1}) = k+3$ for some $i \in [2k]_0$. Then we apply the branching rule (B2) once, we exhaustively apply reduction rules, and again for each instance created by (B2), we introduce a child node of s. The choice of v is not completely arbitrary. If possible, we choose v so that $\text{dist}(v, C) \geq \lceil \frac{d}{2} \rceil$ – note that the cycle C is present in all instances of $\mathcal{R}$.

Case 2: there exists at least one vertex v such that $\mathrm{dist}(v, c_i) = \mathrm{dist}(v, c_{i+1}) = k+3$ for some $i \in [2k]_0$. For every $i \in [2k]_0$, we choose (if exists) a vertex v such that $\mathrm{dist}(v, c_i) = \mathrm{dist}(v, c_{i+1}) = k+3$, and again, if possible, we choose v so that $\mathrm{dist}(v, C) \geq \lceil \frac{d}{2} \rceil$. For each such v, we apply the branching rule (B2) and exhaustively reduction rules, and the children of s are those introduced for all instances created in all applications of (B2).

In both cases, we do not recurse on the children of s for which we applied (B2). Let us analyze the running time of `Recursion Tree` and properties of the tree $\mathcal{R}$.

▶ **Lemma 16.** *Given an instance (G, L) of* LHOM(C_{2k+1}) *with a fixed precolored $(2k+1)$-cycle C and such that $n = |V(G)|$, $\mathrm{diam}(G) \leq d$, the algorithm* `Recursion Tree` *in time $2^{\mathcal{O}((n \log n)^{\frac{d-1}{d}})}$ returns a tree $\mathcal{R}$ whose nodes are labelled with instances of* LHOM(C_{2k+1}) *and (G, L) is a yes-instance if and only if at least one instance corresponding to a leaf of $\mathcal{R}$ is a yes-instance.*

Proof. First we show that for every node s of $\mathcal{R}$ the corresponding instance is a yes-instance if and only if at least one instance corresponding to a child of s is a yes-instance. Let s be a node of $\mathcal{R}$ and let (G', L') be the corresponding instance. The algorithm `Recursion Tree` applies first reduction rules to (G', L') and by Lemma 6, we obtain equivalent instance. Furthermore, we applied to (G', L') either (B1) or (B2) where the branches correspond to all possible colorings of some set of vertices so indeed (G', L') is a yes-instance if and only if at least one instance corresponding to a child of s is a yes-instance. Since the root of $\mathcal{R}$ is labelled with (G, L), we conclude that (G, L) is a yes-instance if and only if at least one instance corresponding to a leaf of $\mathcal{R}$ is a yes-instance.

It remains to analyze the running time. Let $F(\mu)$ be the upper bound on the running time of `Recursion Tree` applied to an instance (G', L') with $\mu = \sum_{\ell=2}^{2k+1} \ell \cdot |V_\ell|$. Let $p(n)$ be a polynomial such that the reduction rules can be exhaustively applied to an instance on n vertices in time $p(n)$ – note that each reduction rule either decreases the number of vertices/sizes of lists or returns NO, so indeed exhaustive application of reduction rules can be performed in polynomial time. Observe that if we apply (B1) to (G', L'), then

$$F(\mu) \leq F\left(\mu - \frac{(\mu \log \mu)^{1/d}}{2k+1}\right) + F(\mu - 1) + 2 \cdot p(n).$$

Indeed, let v be the vertex to which we apply (B1). If there are no lists of type (2) on $N_{G'[V_{\geq 2}]}(v)$, then in the branch where we set $L(v) = \{a\}$, after application of reduction rules, every neighbor of v must have $L(v) \subseteq \{a-1, a+2\}$. If $|L(v)| \geq 2$ and $L(v) \neq \{a-1, a+1\}$, then $|L(v)| \cap \{a-1, a+1\}| < |L(v)|$. Therefore, in this case we decrease sizes of all lists on $N_{G'[V_{\geq 2}]}(v)$. Otherwise, we chose $a \in L(v) \setminus \{j\}$, where $\{j-1, j+1\}$ is the most frequent list of type (2) on $N_{G'[V_{\geq 2}]}(v)$. Since there are exactly $2k+1$ lists of type (2), at least $\frac{1}{2k+1}$-fraction of $N_{G'[V_{\geq 2}]}(v)$ has list of different type than (2) or has list $\{j-1, j+1\}$. Thus, for the branch where we set $L(v) = \{a\}$, the sizes of lists of at least $\frac{1}{2k+1} \cdot (\mu \log \mu)^{1/d}$ vertices decrease. In the branch where we remove a from $L(v)$, we decrease the size of $L(v)$ at least by one. In both branches we apply the reduction rules, so the desired inequality follows.

If we apply (B2) to (G', L') – recall that we stop recursing in this case – and if we are in Case 1, then we obtain

$$F(\mu) \leq (2k+1)^{(\mu \log \mu)^{\frac{d-1}{d}}} \cdot p(n),$$

since we guess the coloring on $N^{\leq d-1}_{G'[V_{\geq 2}]}(v)$ whose size is bounded by $(\mu \log \mu)^{\frac{d-1}{d}}$ (in this case we could not apply (B1) so the degrees in $G'[V_{\geq 2}]$ are bounded by $(\mu \log \mu)^{1/d}$) and the number of possible colors is at most $2k+1$.

In Case 2, we have:

$$F(\mu) \leq (2k+1)^{(2k+1)\cdot(\mu \log \mu)^{\frac{d-1}{d}}} \cdot p(n),$$

where additional $(2k+1)$ in the exponent comes from the fact that we applied (B2) possibly $(2k+1)$ times.

We can conclude that $F(\mu) \leq 2^{\mathcal{O}((\mu \log \mu)^{\frac{d-1}{d}})}$ (see for example [9], proof of Theorem 7) which combined with the inequality $\mu \leq (2k+1)n = \mathcal{O}(n)$ completes the proof. ◀

The following lemma shows that we can solve every instance corresponding to a leaf of $\mathcal{R}$ in polynomial time. We only sketch the proof here – for the full proof see [29].

▶ **Lemma 17** (♠)**.** *Let (G', L') be an instance of* LHOM(C_{2k+1}) *such that* $\text{diam}(G') \leq k+3$ *and let C be a fixed precolored $(2k+1)$-cycle. Assume that we applied algorithm* `Recursion Tree` *to (G', L') and let $\mathcal{R}$ be the resulting recursion tree. Let (G, L) be as instance corresponding to a leaf in $\mathcal{R}$. Then (G, L) can be solved in polynomial time.*

Sketch of proof. By Lemma 10, for every vertex u outside the cycle C, there must be $j \in [2k]_0$ such that $\text{dist}(u, c_j) = \text{dist}(u, c_{j+1})$. Since the reduction rules (R1), (R2) cannot be applied and $\text{diam}(G) \leq k+3$, then that distance is either $k+1$, $k+2$, or $k+3$. In case of $k+1$ or $k+2$ by Lemma 8 have that $L(u) \subseteq \{i-2, i-1, i, i+1, i+2\}$ for some $i \in [2k]_0$, and in case of $k+3$ we have $L(u) \subseteq \{i-3, i-2, i-1, i, i+1, i+2, i+3\}$ for some $i \in [2k]_0$.

Moreover, since for (B2), if we could, we chose vertex v whose distance from C is at least $\lceil \frac{d}{2} \rceil$, each vertex of $V_{\geq 3}$ is at distance $\lfloor \frac{d}{2} \rfloor$ from C. Indeed, every vertex u' that was in $N^{\leq d}_{G[V \geq 2]}(v)$ has list of size at most 2, as we guessed a color either for u' or at least one of its neighbors. So for any vertex u left in $V_{\geq 3}$, the shortest u-v path (whose length is at most the diameter d) should contain a vertex from C and length of that path is at least $\text{dist}(v, C) + \text{dist}(u, C)$. So either all vertices outside C were at distance at most $\lfloor \frac{d}{2} \rfloor$ or v was at distance at least $\lceil \frac{d}{2} \rceil$ and thus u is at distance at most $\lfloor \frac{k+3}{2} \rfloor$ from C. If $k > 3$, then $\lfloor \frac{k+3}{2} \rfloor < k$, which by Lemma 7 implies that $L(u)$ is an independent set. Combining the facts, for $u \in V_{\geq 3}$, $\text{diam} \leq k+2$, and $k > 3$, we obtain $L(u) = \{i-2, i, i+2\}$. By careful analysis we can prove that the same holds in the remaining cases (♠).

Furthermore, observe that if v is a neighbor of u with list $\{i-2, i, i+2\}$ and the reduction rule (R3) cannot be applied, then the possible list of v is $\{i-1, i+1\}$, $\{i-3, i+1\}$, $\{i-1, i+3\}$, $\{i-3, i-1, i+1\}$, or $\{i-1, i+1, i+3\}$. If for some vertex u with list $\{i-2, i, i+2\}$, some of possible lists is not present on $N(u)$, then we add v with such a list to G and make it adjacent to u. Note that now the diameter of G might increase, but we will not care about the diameter anymore. Moreover, any list homomorphism on $G - v$ can be extended to v, whose degree is 1. So since now, we can assume that for u with list $\{i-2, i, i+2\}$, all lists $\{i-1, i+1\}$, $\{i-3, i+1\}$, $\{i-1, i+3\}$ are present on $N(u)$ – this will allow us to encode coloring of u on its neighbors with lists of size 2.

BCSP instance. We start with $\text{BCSP}(C_{2k+1}, G - V_3, L)$. Then, for every vertex $v \in V_3$ and for every $v', v'' \in V_2 \cap N(v)$, we leave in $C(v', v'')$ only such pairs that have a common neighbor in $L(v)$. Furthermore, for every edge uv with $u, v \in V_3$ and lists $L(u) = \{i-2, i, i+2\}$, $L(v) = \{i-1, i+1, i+3\}$, and for every pair $u', v' \in V_2$ such that $uu', vv' \in E(G)$, we remove (if they are present) from $C(u', v')$ the following pairs: $(i-3, i+2)$, $(i-1, i+4)$, and $(i+3, i-2)$. This completes the construction of (V, L, C), which is equivalent to (G, L) (♠).

Clearly (V, L, C) is constructed in polynomial time. Moreover, since all the lists have size at most 2, by Theorem 5, (V, L, C) can be solved in polynomial time, which completes the proof. ◀

Now we are ready to prove Theorem 15.

Proof of Theorem 15. Let (G, L) be an instance of LHOM(C_{2k+1}) such that diam(G) is at most $d \in \{k+2, k+3\}$. As in Theorem 1, first for every $i \in [2k]_0$, we check in polynomial time whether there is a list homomorphism $\varphi : (G, L) \to C_{2k+1}$ such that no vertex is mapped to i – this can be done by [12]. If there is no such list homomorphism, we guess $2k + 1$ vertices $c_0, \ldots, c_{2k}$ which will be colored so that c_i is mapped to i. We add the edges $c_i c_{i+1}$ and we obtain an induced $(2k + 1)$-cycle C (if not, then we are dealing with a no-instance). Note that adding edges cannot increase the diameter and since the edges are added between vertices precolored with consecutive vertices, we obtain an equivalent instance.

Now for (G, L) and C as the fixed precolored $(2k+1)$-cycle we use the algorithm `Recursion Tree`, which by Lemma 16 in time $2^{\mathcal{O}((n \log n)^{\frac{d-1}{d}})}$ returns a tree $\mathcal{R}$. Moreover, in order to solve the instance (G, L) it is enough to solve every instance corresponding to a leaf of $\mathcal{R}$ by Lemma 16, and by Lemma 17, we can solve each such instance in polynomial time. Furthermore, since the size of $\mathcal{R}$ is bounded by the running time, the instance (G, L) can be solved in time $2^{\mathcal{O}((n \log n)^{\frac{d-1}{d}})} \cdot n^{\mathcal{O}(1)} = 2^{\mathcal{O}((n \log n)^{\frac{d-1}{d}})}$, which completes the proof. ◀

6 Beyond odd cycles

In this section we consider target graphs other than odd cycles. Instead, we focus on input graphs with diameter at most 2. Since homomorphisms preserve edges, for graphs G, H, a homomorphism $\varphi : G \to H$ and a sequence of vertices $v_1, \ldots, v_k$ forming a path in G, the sequence $\varphi(v_1), \ldots, \varphi(v_k)$ forms a walk in H. Therefore, if G has diameter at most 2, we can assume that H has also diameter at most 2. The following observation is straightforward.

▶ **Observation 18.** *Let G, H be graphs such that G is connected. If there exists a homomorphism $\varphi : G \to H$, then the image $\varphi(V(G))$ induces in H a subgraph with diameter at most* diam(G)*.*

We prove the following stronger version of Theorem 4.

▶ **Theorem 19.** *Let H be a simple triangle-free graph. Then LHOM(H) is polynomial-time solvable on diameter-2 graphs.*

Proof. Let (G, L) be an instance of LHOM(H). We guess the set of colors that will be used – by Observation 18 they should induce a diameter-2 subgraph H' of H. For each such H', we guess $h' = |H'|$ vertices $v_1, \ldots, v_{h'}$ of G that will be injectively mapped to $V(H') = \{x_1, \ldots, x_{h'}\}$. For each tuple $(H', v_1, \ldots, v_{h'})$, we solve the instance (G, L') of LHOM(H'), where $L'(v) = \{x_i\}$ for $v = v_i$, $i \in [h']$ and $L'(v) = L(v)$ otherwise. Note that (G, L) is a yes-instance if and only if at least one instance (G, L') is a yes-instance.

First for every edge $x_i x_j \in E(H')$, if v_i, v_j are non-adjacent, we add the edge $v_i v_j$ to G – note that this operation is safe, since we cannot increase the diameter by adding edges and we only add edges between vertices that must be mapped to neighbors in H'. Therefore, we can assume that the set $V' = \{v_1, \ldots, v_{h'}\}$ induces a copy of H' in G (if not, then we have an extra edge, which means that we are dealing with a no-instance and we reject immediately). Furthermore, we exhaustively apply reduction rules.

So from now on we assume that the instance (G, L') is reduced. We claim that either (G, L') is a no-instance or $V(G) = \{v_1, \ldots, v_{h'}\}$, i.e., after exhaustive application of the reduction rules the graph G is isomorphic to H'. Note that in the latter case we can return YES as an answer.

Suppose there is $v \in V(G) \setminus V'$. Moreover, we choose such v which is adjacent to some vertex of V' (see Figure 1). Suppose that there exists $\varphi : (G, L') \to H'$ and let $x_i = \varphi(v)$. Then v cannot be adjacent to v_i since there are no loops in H'. Furthermore, the only neighbors of v in V' can be the neighbors of v_i. Suppose that there is $v_j \in N_G(v_i) \cap V'$ which is non-adjacent to v. Since the diameter of G is at most 2, then there must be $u \in N_G(v) \cap N_G(v_j)$. Observe that $u \notin V'$. Indeed, v does not have any neighbors in $V' \setminus N_G(v_i)$ and if $u \in N_G(v)$, then there is a triangle uv_iv_j in a copy of H', a contradiction. Furthermore, it must hold that $\varphi(u)$ is adjacent to x_i in H' as u is adjacent to v and $\varphi(v) = x_i$, and similarly, $\varphi(u)$ must be adjacent to x_j as u is adjacent to v_j. Then $\varphi(u)x_ix_j$ forms a triangle in H', a contradiction. Thus v must be adjacent to all vertices of $N(v_i) \cap V'$.

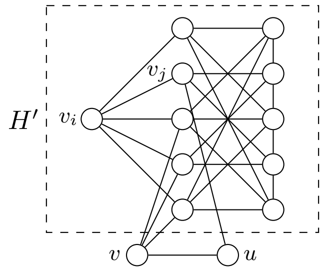

Figure 1 Copy of H' in G and a vertex v such that for some homomorphism φ, it holds $\varphi(v_i) = \varphi(v)$. We show that a vertex u which is a common neighbor of v and some neighbor v_j of v_i in the copy of H' cannot exist.

Since (R3) cannot be applied, each vertex of $L'(v)$ is adjacent to all vertices of $N_H(x_i)$. Moreover, since (R5) cannot be applied, it holds that $L'(v) = \{x_i\}$. Indeed, otherwise there is $x_{i'} \neq x_i$ such that $x_{i'} \in L'(v)$. Recall that $x_{i'}$ is adjacent to all vertices of $N_H(x_i)$. Therefore, $N_H(x_i) \subseteq N_H(x_{i'})$, and thus one of $x_i, x_{i'}$ should have been removed from $L'(v)$ by (R5). Furthermore, since (R6) cannot be applied, we must have $v = v_i \in V'$, a contradiction. This completes the proof. ◀

7 Conclusion

In this paper we studied the computational complexity of Hom(C_{2k+1}) problem on bounded-diameter graphs. We proved that for $k \geq 2$, the Hom(C_{2k+1}) problem can be solved in polynomial-time on diameter-$(k+1)$ graphs and we gave subexponential-time algorithms for diameter-$(k+2)$ and -$(k+3)$ graphs. We also proved that Hom(H) for triangle-free graph H, can be solved in polynomial time on diameter-2 graphs.

The main open problem in this area remains the question whether 3-Coloring on diameter-2 graphs can be solved in polynomial time. However, as more reachable, we propose the following future research directions. (i) Is the Hom(C_{2k+1}) problem NP-hard for the subexponential-time cases, i.e., diameter-$(k+2)$ or -$(k+3)$ graphs? (2) Let H be a diameter-2 graph such that H contains a triangle but $H \not\to K_3$. Is the Hom(H) problem NP-hard?

References

1 Laurent Beaudou, Florent Foucaud, and Reza Naserasr. Smallest c_{2l+1}-critical graphs of odd-girth 2k+1. *Discret. Appl. Math.*, 319:564–575, 2022. doi:10.1016/J.DAM.2021.08.040.

2 Marthe Bonamy, Konrad K. Dabrowski, Carl Feghali, Matthew Johnson, and Daniël Paulusma. Independent feedback vertex sets for graphs of bounded diameter. *Inf. Process. Lett.*, 131:26–32, 2018. doi:10.1016/J.IPL.2017.11.004.

3 Christoph Brause, Petr A. Golovach, Barnaby Martin, Pascal Ochem, Daniël Paulusma, and Siani Smith. Acyclic, star, and injective colouring: Bounding the diameter. *Electron. J. Comb.*, 29(2), 2022. doi:10.37236/10738.

4 Andrei A. Bulatov and Amirhossein Kazeminia. Complexity classification of counting graph homomorphisms modulo a prime number. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 – 24, 2022*, pages 1024–1037. ACM, 2022. doi:10.1145/3519935.3520075.

5 Jin-Yi Cai and Ashwin Maran. The complexity of counting planar graph homomorphisms of domain size 3. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1285–1297. ACM, 2023. doi:10.1145/3564246.3585173.

6 Victor A. Campos, Guilherme de C. M. Gomes, Allen Ibiapina, Raul Lopes, Ignasi Sau, and Ana Silva. Coloring problems on bipartite graphs of small diameter. *Electron. J. Comb.*, 28(2):2, 2021. doi:10.37236/9931.

7 Rajesh Chitnis, László Egri, and Dániel Marx. List h-coloring a graph by removing few vertices. *Algorithmica*, 78(1):110–146, 2017. doi:10.1007/S00453-016-0139-6.

8 Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 210–223. ACM, 2017. doi:10.1145/3055399.3055502.

9 Michal Dębski, Marta Piecyk, and Paweł Rzążewski. Faster 3-coloring of small-diameter graphs. *SIAM J. Discret. Math.*, 36(3):2205–2224, 2022. doi:10.1137/21M1447714.

10 Oliver Ebsen and Mathias Schacht. Homomorphism thresholds for odd cycles. *Comb.*, 40(1):39–62, 2020. doi:10.1007/S00493-019-3920-8.

11 Keith Edwards. The complexity of colouring problems on dense graphs. *Theor. Comput. Sci.*, 43:337–343, 1986. doi:10.1016/0304-3975(86)90184-2.

12 Tomás Feder, Pavol Hell, and Jing Huang. List homomorphisms and circular arc graphs. *Comb.*, 19(4):487–505, 1999. doi:10.1007/S004939970003.

13 Tomás Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *J. Graph Theory*, 42(1):61–80, 2003. doi:10.1002/JGT.10073.

14 Jacob Focke, Dániel Marx, and Paweł Rzążewski. Counting list homomorphisms from graphs of bounded treewidth: tight complexity bounds. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 – 12, 2022*, pages 431–458. SIAM, 2022. doi:10.1137/1.9781611977073.22.

15 Robert Ganian, Thekla Hamm, Viktoriia Korchemna, Karolina Okrasa, and Kirill Simonov. The fine-grained complexity of graph homomorphism parameterized by clique-width. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 66:1–66:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.66.

16 A. M. H. Gerards. Homomorphisms of graphs into odd cycles. *J. Graph Theory*, 12(1):73–83, 1988. doi:10.1002/JGT.3190120108.

17 Carla Groenland, Isja Mannens, Jesper Nederlof, Marta Piecyk, and Paweł Rzążewski. Towards tight bounds for the graph homomorphism problem parameterized by cutwidth via asymptotic rank parameters. *CoRR*, abs/2312.03859, 2023. doi:10.48550/arXiv.2312.03859.

18 Pavol Hell and Jaroslav Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990. doi:10.1016/0095-8956(90)90132-J.

19 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/JCSS.2000.1727.

20 Tereza Klimosová and Vibha Sahlot. 3-coloring c_4 or c_3-free diameter two graphs. In Pat Morin and Subhash Suri, editors, *Algorithms and Data Structures – 18th International Symposium, WADS 2023, Montreal, QC, Canada, July 31 – August 2, 2023, Proceedings*, volume 14079 of *Lecture Notes in Computer Science*, pages 547–560. Springer, 2023. `doi:10.1007/978-3-031-38906-1_36`.

21 Aleksej Dmitrievich Korshunov. On the diameter of graphs. *Soviet Math*, 12:302:305, 1971.

22 Hong-Jian Lai. Unique graph homomorphisms onto odd cycles, II. *J. Comb. Theory, Ser. B*, 46(3):363–376, 1989. `doi:10.1016/0095-8956(89)90056-7`.

23 Barnaby Martin, Daniël Paulusma, and Siani Smith. Colouring h-free graphs of bounded diameter. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 14:1–14:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPICS.MFCS.2019.14`.

24 Barnaby Martin, Daniël Paulusma, and Siani Smith. Colouring graphs of bounded diameter in the absence of small cycles. *Discret. Appl. Math.*, 314:150–161, 2022. `doi:10.1016/J.DAM.2022.02.026`.

25 George B. Mertzios and Paul G. Spirakis. Algorithms and almost tight results for 3-colorability of small diameter graphs. *Algorithmica*, 74(1):385–414, 2016. `doi:10.1007/S00453-014-9949-6`.

26 Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 74:1–74:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPICS.ESA.2020.74`.

27 Karolina Okrasa and Paweł Rzążewski. Fine-grained complexity of the graph homomorphism problem for bounded-treewidth graphs. *SIAM J. Comput.*, 50(2):487–508, 2021. `doi:10.1137/20M1320146`.

28 Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

29 Marta Piecyk. C_{2k+1}-coloring of bounded-diameter graphs. *CoRR*, abs/2403.06694, 2024. `doi:10.48550/arXiv.2403.06694`.

30 Sebastian Schnettler. A structured overview of 50 years of small-world research. *Soc. Networks*, 31(3):165–178, 2009. `doi:10.1016/J.SOCNET.2008.12.004`.

31 Petra Sparl and Janez Zerovnik. Homomorphisms of hexagonal graphs to odd cycles. *Discret. Math.*, 283(1-3):273–277, 2004. `doi:10.1016/J.DISC.2003.12.012`.