Approximability and Inapproximability of Minimum Cost Homomorphism *

Pavol Hell † Monaldo Mastrolilli ‡ Mayssam Nevisi § Akbar Rafiey ¶ Arash Rafiey ‖

Abstract

We study the approximability and hardness of approximation of minimum cost homomorphism to target graph H, MinHOM(H). When H is a bipartite graph, we prove that if H is a co-circular arc bigraph, then MinHOM(H) admits a polynomial time constant ratio approximation algorithm; otherwise, MinHOM(H) is known to be not approximable. For the purposes of the approximation, we provide a new characterization of co-circular arc bigraphs by the existence of min ordering. Our algorithm is then obtained by derandomizing a two-phase randomized procedure.

Moreover, we provide a complete classification of approximable cases of graphs. That is, we prove MinHOM(H) has a constant factor approximation algorithm if graph H is a bi-arc graph (i.e., admits a conservative majority polymorphism), otherwise, it is inapproximable assuming P≠NP;

Finally, we complement our positive results with hardness of approximation results for graphs. We show that MinHOM(H) is 1.128-approx-hard and 1.242-UGC-hard. Thus, we obtain a dichotomy theorem for approximability and inapproximability of MinHOM(H) when H is a graph.

1 Introduction

We study the approximability of the minimum cost homomorphism problem, introduced below. A *c-approximation algorithm* produces a solution of cost at most c times the minimum

*An extended abstract of the approximation part has appeared in [16, 31]

†Simon Fraser University, Burnaby, BC, Canada, pavol@sfu.ca

‡IDSIA, Manno-Lugano, Swtizerland, monaldo@idsia.ch supported by the Swiss National Science Foundation project 200020-122110/1,"Approximation Algorithms for Machine Scheduling II"

§Microsoft, Vancouver, Canada, mayssam@gmail.com

¶University of California San Diego, San Diego, CA, USA, arafiey@ucsd.edu

‖Indiana State University, IN, USA, arash.rafiey@indstate.edu supported by NSF 1751765, and by the Swiss National Science Foundation project 200020-122110/1,"Approximation Algorithms for Machine Scheduling II"

cost. A *constant ratio* approximation algorithm is a c-approximation algorithm for some constant c. When we say a problem has a c-approximation algorithm, we mean a polynomial-time algorithm. We say that a problem is *not approximable* if there is no polynomial-time approximation algorithm with a multiplicative guarantee unless $P = NP$.

The minimum cost homomorphism problem was introduced in [12]. It consists of minimizing a certain cost function over all homomorphisms from an input graph G to a fixed graph H. This offers a natural and practical way to model many optimization problems. For instance, in [12] it was used to model a problem of minimizing the cost of a repair and maintenance schedule for large machinery. It generalizes many other problems such as list homomorphism problems (see below), and various optimum cost chromatic partition problems [13, 22, 23, 27]. (A different kind of the minimum cost homomorphism problem was introduced in [1].) Certain minimum cost homomorphism problems have polynomial-time algorithms [10, 11, 12], but most are NP-hard. Therefore we investigate the approximability of these problems. Note that we approximate the cost over real homomophisms, rather than approximating the maximum weight of satisfied constraints, as in, say, MAXSAT.

We call a graph *reflexive* if every vertex has a loop, and *irreflexive* if no vertex has a loop. An *interval graph* is a graph that is the intersection graph of a family of real intervals, and a *circular arc graph* is a graph that is the intersection graph of a family of arcs on a circle. We interpret the concept of an intersection graph literally, thus any intersection graph is automatically reflexive, since a set always intersects itself. A bipartite graph whose complement is a circular arc graph, will be called a *co-circular arc bigraph.* When forming the complement, we take all edges that were not in the graph, including loops and edges between vertices in the same color. In general, the word *bigraph* will be reserved for a bipartite graph with a fixed bipartition of vertices; we shall refer to *white* and *black* vertices to reflect this fixed bipartition. Bigraphs can be conveniently viewed as directed bipartite graphs with all edges oriented from the white to the black vertices. Thus, by definition, interval graphs are reflexive, and co-circular arc bigraphs are irreflexive. Despite the apparent differences in their definition, these two graph classes exhibit certain natural similarities [6, 7]. There is also a concept of an *interval bigraph* H, which is defined for two families of real intervals, one family for the white vertices and one family for the black vertices: a white vertex is adjacent to a black vertex if and only if their corresponding intervals intersect. Interval bigraphs, have been studied in [14, 29, 30].

A reflexive graph is a *proper interval graph* if it is an interval graph in which the defining family of real intervals can be chosen to be inclusion-free. A bigraph is a *proper interval bigraph* if it is an interval bigraph in which the defining two families of real intervals can be chosen to be inclusion-free. It turns out [14] that proper interval bigraphs are a subclass of co-circular arc bigraphs.

A *homomorphism* of a graph G to a graph H is a mapping $f : V(G) \to V(H)$ such that for any edge xy of G the pair $f(x)f(y)$ is an edge of H.

Let H be a fixed graph. The *list homomorphism problem* to H, denoted ListHOM(H), seeks, for a given input graph G and lists $L(x) \subseteq V(H), x \in V(G)$, a homomorphism f of G to H such that $f(x) \in L(x)$ for all $x \in V(G)$. It was proved in [7] that for irreflexive graphs, the problem ListHOM(H) is polynomial-time solvable if H is a co-circular arc bigraph, and is NP-complete otherwise. It was shown in [6] that for reflexive graphs H, the problem ListHOM(H) is polynomial-time solvable if H is an interval graph, and is NP-complete otherwise.

The *minimum cost homomorphism problem* to H, denoted MinHOM(H), seeks, for a given input graph G and vertex-mapping costs $c(x, u), x \in V(G), u \in V(H)$, a homomorphism f of G to H that minimizes total cost $\sum_{x \in V(G)} c(x, f(x))$.

As mentioned above the MinHOM problem offers a natural and practical way to model and generalizes many optimization problems.

Example 1.1 (Vertex Cover). *This problem can be seen as MinHOM(H) where $V(H) = \{a, b\}$, $E(H) = \{aa, ab\}$, and $c(u, a) = 1$, $c(u, b) = 0$ for every vertex $u \in G$.*

Example 1.2 (Chromatic Sum). *In this problem, we are given a graph G, and the objective is to find a proper coloring of G with colors $\{1, \ldots, k\}$ with* minimum color sum. *This can be seen as MinHOM where H is a clique of size k with $V(H) = \{1, \ldots, k\}$ and the cost function is defined as $c(u, i) = i$. The Chromatic Sum problem appears in many applications such as resource allocation problems [3].*

Example 1.3 (Multiway Cut). *Let G be a graph where each edge e has a non-negative weight $w(e)$. There are also k specific (terminal) vertices, $s_1, s_2, \ldots, s_k$ of G. The goal is to partition the vertices of G into k parts so that each part $i \in \{1, 2, \ldots, k\}$, contains s_i and the sum of the weights of the edges between different parts is minimized. Let H be a graph with vertex set $\{a_1, a_2, \ldots, a_k\} \cup \{b_{i,j} \mid 1 \leq i < j \leq k\}$. The edge set of H is $\{a_i a_i, a_i b_{i,j}, b_{i,j} a_j, a_j a_j \mid 1 \leq i < j \leq k\}$. Now obtain the graph G' from G by replacing every edge $e = uv$ of G with the edges $ux_e, x_e v$ where x_e is a new vertex. The cost function c is as follows. $c(s_i, a_i) = 0$, else $c(s_i, d) = |G|$ for $d \neq a_i$. For every $u \in G \setminus \{s_1, s_2, \ldots, s_k\}$, set $c(u, s_i) = 0$. Set $c(x_e, b_{i,j}) = w(e)$. Now, finding a minimum multiway cut in G is equivalent to finding a minimum-cost homomorphism from graph G' to H.*

The complexity of MinHOM(H) for graphs and digraphs have been well-understood [11, 20].It was proved in [11] that for irreflexive graphs, the problem MinHOM(H) is polynomial-time solvable if H is a proper interval bigraph, and it is NP-complete otherwise. It was also shown there that for reflexive graphs H, the problem MinHOM(H) is polynomial time solvable if H is a proper interval graph, and it is NP-complete otherwise.

In [28], the authors have shown that MinHOM(H) is not approximable if H is a graph that is not bipartite or not a co-circular arc graph, and gave a randomized 2-approximation algorithms for MinHOM(H) for a certain subclass of co-circular arc bigraphs H. The authors have asked for the exact complexity classification for these problems. We answer the question by showing that the problem MinHOM(H) in fact has a $|V(H)|$-approximation algorithm for **all** co-circular arc bigraphs H. Thus for an irreflexive graph H the problem MinHOM(H) has a constant ratio approximation algorithm if H is a co-circular arc bigraph, and is not approximable otherwise. We also prove that for a reflexive graph H the problem MinHOM(H) has a constant ratio approximation algorithm if H is an interval graph, and is not approximable otherwise. We use the method of randomized rounding, a novel technique of randomized shifting, and then a simple derandomization.

A *min ordering* of a graph H is an ordering of its vertices $a_1, a_2, \ldots, a_n$, so that the existence of the edges $a_i a_j, a_{i'} a_{j'}$ with $i < i'$ and $j' < j$ implies the existence of the edge $a_i a_{j'}$. A *min-max ordering* of a graph H is an ordering of its vertices $a_1, a_2, \ldots, a_n$, so that the existence of the edges $a_i a_j, a_{i'} a_{j'}$ with $i < i'$ and $j' < j$ implies the existence of the edges $a_i a_{j'}, a_{i'} a_j$. For bigraphs, it is more convenient to speak of two orderings, and we define a *min ordering* of a bigraph H to be an ordering $a_1, a_2, \ldots, a_p$ of the white vertices and an ordering $b_1, b_2, \ldots, b_q$ of the black vertices, so that the existence of the edges $a_i b_j, a_{i'} b_{j'}$ with $i < i', j' < j$ implies the existence of the edge $a_i b_{j'}$; and a *min-max ordering* of a bigraph H to be an ordering of $a_1, a_2, \ldots, a_p$ of the white vertices and an ordering $b_1, b_2, \ldots, b_q$ of the black vertices, so that the existence of the edges $a_i b_j, a_{i'} b_{j'}$ with $i < i', j' < j$ implies the existence of the edges $a_i b_{j'}, a_{i'} b_j$. (Both are instances of a general definition of min ordering for directed graphs [19].)

In Section 2 we prove that co-circular arc bigraphs are precisely the bigraphs that admit a min ordering. In the realm of reflexive graphs, such a result is known about the class of interval graphs (they are precisely the reflexive graphs that admit a min ordering) [18].

Approximability results. In Section 3 we recall that MinHOM(H) is not approximable when H does not have min ordering, and describe a $|V(H)|$-approximation algorithm when H is a bigraph that admits a min ordering. In Section 4, we further apply our technique for graphs (vertices with possible loops) and show that when H is a bi-arc graph then MinHOM(H) has a $2|V(H)|$-approximation algorithm. Note that, for graphs, MinHOM(H) is not approximable if H is not a bi-arc graph. Hence, our result gives a dichotomy classification for approximation of MinHOM(H) when H is a graph.

Inapproximability results. As pointed out, the MinHOM(H) is not approximable if ListHOM(H) is not polynomial-time solvable. This rules out the possibility of having an approximation algorithm for graphs that are not bi-arc. However, there are no known inapproximability results for the cases where MinHOM(H) is NP-complete. We, therefore, complete the picture by considering a much bigger class of graphs than bi-arc graphs and present inapproximability results for them. That is the class of graphs for which MinHOM is NP-complete. This class of graphs has been characterized in [11] and are known as graphs that do not admit a *min-max ordering*. The obstructions for min-max ordering for graphs and digraphs have been provided in [21]. This characterization was used to show the NP-completeness of MinHOM together with the NP-completeness of the maximum independent set problem [20]. However, in this paper, we must develop an array of approximation-preserving reductions to obtain our inapproximability results.

2 Co-circular bigraphs and min ordering

A reflexive graph has a min ordering if and only if it is an interval graph [18]. In this section we prove a similar result about bigraphs. Two auxiliary concepts from [7, 9] are introduced first.

An *edge asteroid* of a bigraph H consists of $2k+1$ disjoint edges $a_0b_0, a_1b_1, \dots, a_{2k}b_{2k}$ such that each pair a_i, a_{i+1} is joined by a path disjoint from all neighbours of $a_{i+k+1}b_{i+k+1}$ (subscripts modulo $2k+1$).

An *invertible pair* in a bigraph H is a pair of white vertices a, a' and two pairs of walks $a = v_1, v_2, \dots, v_k = a'$, $a' = v'_1, v'_2, \dots, v'_k = a$, and $a' = w_1, w_2, \dots, w_m = a$, $a = w'_1, w'_2, \dots, w'_m = a'$ such that v_i is not adjacent to v'_{i+1} for all $i = 1, 2, \dots, k$ and w_j is not adjacent to w'_{j+1} for all $j = 1, 2, \dots, m$.

Theorem 2.1. *A bigraph H is a co-circular arc graph if and only if it admits a min ordering.*

Proof. Consider the following statements for a bigraph H:

1. H has no induced cycles of length greater than three and no edge asteroids
2. H is a co-circular-arc graph
3. H has a min ordering
4. H has no invertible pairs

$1 \Rightarrow 2$ is proved in [7].

$2 \Rightarrow 3$ is seen as follows: Suppose H is a co-circular arc bigraph; thus the complement $\overline{H}$ is a circular arc graph that can be covered by two cliques. It is known for such graphs that there exist two points, the *north pole* and the *south pole*, on the circle, so that the white vertices u of H correspond to arcs A_u containing the north pole but not the south pole, and the black vertices v of H correspond to arcs A_v containing the south pole but not the north pole. We now define a min ordering of H as follows. The white vertices are ordered according to the clockwise order of the corresponding clockwise extremes, i.e., u comes before u' if the clockwise end of A_u precedes the clockwise end of $A_{u'}$. The same definition, applied to the black vertices v and arcs A_v, gives an ordering of the black vertices of H. It is now easy to see from the definitions that if $uv, u'v'$ are edges of H with $u < u'$ and $v > v'$, then A_u and $A_{v'}$ must be disjoint, and so uv' is an edge of H.

$3 \Rightarrow 4$ is easy to see from the definitions (see, for instance [9]).

$4 \Rightarrow 1$ is checked as follows: If C is an induced cycle in H, then C must be even, and any two of its opposite vertices together with the walks around the cycle form an invertible pair of H. In an edge-asteroid $a_0b_0, \dots, a_{2k}b_{2k}$ as defined above, it is easy to see that, say, a_0, a_k is an invertible pair. Indeed, there is, for any i, a walk from a_i to a_{i+1} that has no edges to the walk $a_{i+k}, b_{i+k}, a_{i+k}, b_{i+k}, \dots, a_{i+k}$ of the same length. Similarly, a walk $a_{i+1}, b_{i+1}, a_{i+1}, b_{i+1}, \dots, a_{i+1}$ has no edges to a walk from a_{i+k} to a_{i+k+1} implied by the definition of an edge-asteroid. By composing such walks we see that a_0, a_k is an invertible pair. □

We note that it can be decided in time polynomial in the size of H, whether a graph H is a (co-)circular arc bigraph [15].

3 Approximation of MinHOM for bipartite graphs

In this section we describe our approximation algorithm for MinHOM(H) in the case the fixed bigraph H has a min ordering, i.e., is a co-circular arc bigraph, cf. Theorem 2.1. We recall that if H is not a co-circular arc bigraph, then the list homomorphism problem ListHOM(H) is NP-complete [7], and this implies that MinHOM(H) is not approximable for such graphs H [28]. By Theorem 2.1 we conclude the following.

Theorem 3.1. *If a bigraph H has no min ordering, then MinHOM(H) is not approximable.*

Our main result is the following converse: if H has a min ordering (is a co-circular arc bigraph), then there exists a constant ratio approximation algorithm (since H is fixed, $|V(H)|$ is a constant.).

Theorem 3.2. *If H is a bigraph that admits a min ordering, then MinHOM(H) has a $|V(H)|$-approximation algorithm.*

To prove the above theorem we first design an approximation algorithm.

Fixing a min ordering for H. Suppose H has a min ordering with the white vertices ordered $a_1, a_2, \cdots, a_p$, and the black vertices ordered $b_1, b_2, \cdots, b_q$. For every $1 \leq i \leq p$, let $r(i)$ be the first subscript that $a_i b_{r(i)}$ is an edge of H. For every $1 \leq i \leq q$, let $\ell(i)$ be the first subscript that $a_{\ell(i)} b_i$ is an edge of H.

Definition 3.3 (H' and E' construction)**.** *Let E' denote the set of all pairs $a_i b_j$ such that $a_i b_j$ is not an edge of H, but there is an edge $a_i b_{j'}$ of H with $j' < j$ and an edge $a_{i'} b_j$ of H with $i' < i$. Define H' to be the graph with vertex set $V(H)$ and edge set $E(H) \cup E'$. (Note that $E(H)$ and E' are disjoint sets.)*

Observation 3.4. *The ordering $a_1, a_2, \cdots, a_p$, and $b_1, b_2, \cdots, b_q$ is a min-max ordering of H'.*

Proof. We show that for every pair of edges $e = a_i b_{j'}$ and $e' = a_{i'} b_j$ in $E(H) \cup E'$, with $i' < i$ and $j' < j$, both $f = a_i b_j$ and $f' = a_{i'} b_{j'}$ are in $E(H) \cup E'$. If both e and e' are in $E(H)$, $f \in E(H) \cup E'$ and $f' \in E(H)$. If one of the edges, say e, is in E', there is a vertex $b_{j''}$ with $a_i b_{j''} \in E(H)$ and $j'' < j'$, and a vertex $a_{i''}$ with $a_{i''} b_{j'} \in E(H)$ and $i'' < i$. Now, $a_{i'} b_j$ and $a_i b_{j''}$ are both in $E(H)$, so $f \in E(H) \cup E'$. We may assume that $i'' \neq i'$, otherwise $f' = a_{i''} b_{j'} \in E(H)$. If $i'' < i'$, then $f' \in E(H) \cup E'$ because $a_{i'} b_{j''} \in E(H)$; and if $i'' > i'$, then $f' \in E(H)$ because $a_{i'} b_j \in E(H)$.

If both edges e, e' are in E', then the earlier neighbours of a_i and b_j in $E(H)$ imply that $f \in E(H) \cup E'$, and the earlier neighbours of $a_{i'}$ and $b_{j'}$ in $E(H)$ imply that $f' \in E(H) \cup E'$. □

Observation 3.5. *Let $e = a_i b_j \in E'$. Then a_i is not adjacent in $E(H)$ to any vertex after b_j, or b_j is not adjacent in $E(H)$ to any vertex after a_i.*

Proof. This easily follows from the fact that $a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_q$ is a min ordering. □

Minimize $\sum_{u\in U, i\in[p]} c(u,a_i)(x_{u,a_i} - x_{u,a_{i+1}}) + \sum_{v\in V, j\in[q]} c(v,b_j)(x_{v,b_i} - x_{v,b_{j+1}})$		
Subject to:		
$0 \le x_{u,a_i}, v_{v,b_j} \le 1$	$\forall u,v \in V(G), a_i, b_j \in V(H)$	(C1)
$x_{u,a_1} = x_{v,b_1} = 1$ and $x_{u,a_{p+1}} = x_{v,b_{q+1}} = 0$		(C2)
$x_{v,b_{i+1}} \le x_{v,b_i}$ and $x_{u,a_{i+1}} \le x_{u,a_i}$	$\forall v \in V, u \in U, a_i, b_i \in V(H)$	(C3)
$x_{u,a_i} \le x_{v,b_{r(i)}}$ and $x_{v,b_i} \le x_{u,a_{\ell(i)}}$	$\forall uv \in E(G)$	(C4)
$x_{v,b_j} \le x_{u,a_s} + \sum_{a_t b_j \in E(H), t<i} (x_{u,a_t} - x_{u,a_{t+1}})$	$\forall uv \in E(G), a_i b_j \in E'$, a_s is the first neighbor of b_j after a_i	(C5)
$x_{u,a_i} \le x_{v,b_s} + \sum_{a_i b_t \in E(H), t<j} (x_{v,b_t} - x_{v,b_{t+1}})$	$\forall uv \in E(G), a_i b_j \in E'$ b_s is the first neighbor of a_i after b_j	(C6)
$x_{u,a_i} - x_{u,a_{i+1}} \le \sum_{a_i b_t \in E(H), t<j} (x_{v,b_t} - x_{v,b_{t+1}})$	$\forall uv \in E(G)$, $a_i b_j \in E'$, and a_i has no neighbor after b_j	(C7)
$x_{v,b_j} - x_{v,b_{j+1}} \le \sum_{a_t b_j \in E(H), t<i} (x_{u,a_t} - x_{u,a_{t+1}})$	$\forall uv \in E(G)$, $a_i b_j \in E'$, and b_j has no neighbor after a_i	(C8)

Table 1: Linear program $\mathcal{S}$

Assumption about the input and introducing the variables. First we assume input bipartite graph $G = (U, V)$ is connected, as otherwise, we solve the problem for each connected component of G. Here U represent the left vertices of G and V represent the right vertices of G. We further look for a homomorphism f that maps vertices U to $\{a_1, a_2, \dots, a_p\}$ and vertices V to $\{b_1, b_2, \dots, b_p\}$.

For every vertex $u \in U$, and every a_i, define the variable x_{u,a_i}, and for every $v \in V$ and b_j, define the variable x_{v,b_j}.

System of linear equations S. Having defined the variables x_{u,a_i}, x_{v,b_j}, we introduce the linear program $\mathcal{S}$ shown in table 1 that formulates MinHOM(H). The intuition is if variable $x_{u,a_i} = 1$ and $x_{u,a_{i+1}} = 0$, then we map u to a_i. Thus, we add constraint (C3) that has inequalities $x_{u,a_{i+1}} \le x_{u,a_i}$ and $x_{v,a_{j+1}} \le x_{v,a_j}$. Now, from constraint (C3) and the min ordering, we add constraint (C4). Constraints (C5,C6) are the most important constraints capturing the min ordering property. Using Observation 3.5, constraint (C7,C8) are added to make sure that if we map $u \in U$ ($v \in V$) to a_i (b_j) then the neighbor of u (v), say v (u) is mapped to a neighbor of a_i (b_j).

Lemma 3.6. *If H admits a min-ordering then there is a one to one correspondence between homomorphisms of G to H and the integer solutions of $\mathcal{S}$.*

Proof. Suppose f is a homomorphism from G to H. If $f(u) = a_i$ then set $x_{u,a_j} = 1$, for all $j \le i$ and $x_{u,a_j} = 0$ for all $j > i$. Similar treatment for v and b_j. Clearly, constraints $C1, C2, C3$, and $C4$ are satisfied. Now for all u and v in G with $f(u) = a_i$ and $f(v) = b_j$ we have that $x_{u,a_i} - x_{u,a_{i+1}} = x_{v,b_j} - x_{v,b_{j+1}} = 1$. Moreover, since f is a homomorphism constraint (C7,C8) are also satisfied.

We show that constraint (C5) holds. For, contradiction, assume that the inequality in (C5) fails. This means that for some edge uv of G and some arc $a_ib_j \in E'$, we have $x_{v,b_j} = 1$, $x_{u,a_s} = 0$, and the sum of $(x_{u,a_t} - x_{u,a_{t+1}})$, over all $t < i$ such that a_t is a neighbor of a_j, is zero. The latter two facts easily imply that $f(u) = a_i$. Since b_j has a neighbor after a_i, Observation 3.5 tells us that a_i has no neighbor after b_j and $x_{v,b_{j+1}} = 0$, whence $f(v) = b_j$ and thus $a_ib_j \in E(H)$, a contradiction the assumption $a_ib_j \in E'$. By a similar argument (C6) is also satisfied.

Conversely, from an integer solution for $\mathcal{S}$, we define a mapping f from G to H as follows. For every $u \in U$, set $f(u) = a_i$ when i is the largest subscript with $x_{u,a_i} = 1$. Similarly, for every $v \in V$ set $f(v) = b_j$ when j is the largest subscript with $x_{v,b_j} = 1$.

Let uv be an edge of G and assume $f(u) = a_i$, $f(v) = b_j$. Note that $x_{u,a_i} - x_{u,a_{i+1}} = x_{v,b_j} - x_{v,b_{j+1}} = 1$ and for all other t we have $x_{v,b_t} - x_{v,b_{t+1}} = 0$. If a_ib_j is an edge of H we are done. Suppose this is not the case. Since constraints C4 is satisfied, a_i has a neighbor before b_j and b_j has a neighbor before a_i Thus, $a_ib_j \in E'$. First suppose a_i has no neighbor after b_j. Now, $0 = \sum_{a_ib_t \in E(H), t<j} (x_{v,b_t} - x_{v,b_{t+1}})$, violating constraint (C7). Thus, assume a_i has a neighbor after b_j. Now $x_{u,a_i} = 1$, while $x_{v,b_s} = 0$, and for every $t < j$, $x_{v,b_t} - x_{v,b_{t+1}} = 0$, and hence, constraint (C6) is not satisfied, a contradiction. □

Overview of the rounding procedure. Our algorithm will minimize the cost function over $\mathcal{S}$ in polynomial time using a linear programming algorithm. This will generally result in a fractional solution. We will obtain an integer solution by a randomized procedure called *rounding*. We choose a random variable $X \in [0, 1]$, and define the rounded values $\chi_{u,a_i} = 1$ when $x_{u,a_i} \geq X$, and $\chi_{u,a_i} = 0$ otherwise; and similarly define the rounded value χ_{v,b_j} from x_{v,b_j}. Now set $f(u) = a_i$ where $\chi_{u,a_i} = 1$, $\chi_{u,a_{i+1}} = 0$ and set $f(v) = b_j$ where $\chi_{v,b_j} = 1$, $\chi_{v,b_{j+1}} = 0$. In Lemma 3.7 we show that the mapping f is a homomorphism from G to H'. However, f may not be a homomorphism from G to H. Now the algorithm will once more modify the solution f to become a homomorphism of G to H, i.e., to avoid mapping edges of G to the edges in E'. This will be accomplished by another randomized procedure, which we call *shifting*. We choose another random variable $Y \in [0, 1]$, which will guide the shifting. Let F denote the set of all edges in E' to which some edge of G is mapped by f. We also let $F(G) = \{(u, v, f(u), f(v)) | uv \in E(G), f(u)f(v) \in E'\}$.

If F is empty, we need no shifting. Otherwise, let a_ib_j be an edge of F with maximum sum $i + j$ (among all edges of F). By the maximality of $i + j$, we know that a_ib_j is the last edge of F from both a_i and b_j. Now we consider, one by one, $(u, v, a_i, b_j) \in F(G)$ (i.e. $uv \in E(G)$) where $f(u) = a_i$ and $f(v) = b_j$. Since $F \subseteq E'$, by Observation 3.5 either a_i has no neighbor after b_j or b_j has no neighbor after a_i.

Suppose $f(u) = a_i$ and a_i have no neighbor after b_j (the other case is where $f(v) = b_j$ and b_j has no neighbor after a_i). For such a vertex u, consider the set of all vertices a_t with $t < i$ such that $a_tb_j \in E(H)$. This set is not empty, since e is in E' because of two edges of $E(H)$. Suppose the set consists of a_t with subscripts t ordered as $t_1 < t_2 < \dots t_k$. The algorithm now selects one vertex from this set as follows. Let $P_{u,t} = \frac{x_{u,a_t} - x_{u,a_{t+1}}}{P_u}$, where

$$P_u = \sum_{a_t b_j \in E(H),\ t<i} (x_{u,a_t} - x_{u,a_{t+1}}).$$

Then a_{t_q} is selected if $\sum_{p=1}^{q} P_{u,t_p} < Y \leq \sum_{p=1}^{q+1} P_{u,t_p}$. Thus, a concrete a_t is selected with probability $P_{u,t}$, which is proportional to the difference of the fractional values $x_{u,a_t} - x_{u,a_{t+1}}$.

When the selected vertex is a_t, we shift the image of the vertex u from a_i to a_t. This modifies the homomorphism f, and hence the corresponding values of the variables. Namely, $\chi_{u,a_{t+1}}, \ldots, \chi_{u,a_i}$ are reset to 0, keeping all other values the same. Note that the modified mapping is still a homomorphism from G to H'.

We repeat the same process for the next u with these properties, until $a_i b_j$ is no longer in F (because no edge of G maps to it). This ends the iteration on $a_i b_j$, and we proceed to the next edge $a_{i'} b_{j'}$ with maximum $i' + j'$ for the next iteration. Each iteration involves at most $|V(G)|$ shifts. After at most $|E'|$ iterations, the set F is empty and no shift is needed.

It is easy to see, due to min ordering, if the image of vertex u changes because of edge uv with $f(u)f(v) \notin E(H)$, while $f(u)f(w) \in E(H)$ for some other neighbor w of u, by changing the image of u there is no need to change the image of w. We also show that the image of every vertex w in G changes at most once. More details are provided at the beginning of Lemma 3.8.

Algorithm 2 Procedures Shift-Left and Shift-Right

procedure Shift-Left(f, u, v, a_i, b_j, Y)
  Let $a_{t_1}, a_{t_2}, \ldots, a_{t_k}$ be the neighbors of b_j in H before a_i
  Let $P_u \leftarrow \sum_{l=1}^{k} (x_{u,a_{t_l}} - x_{u,a_{t_l+1}})$, and let $P_{u,a_{t_q}} \leftarrow \sum_{l=1}^{q} (x_{u,a_{t_l}} - x_{u,a_{t_l+1}})/P_u$
  if $P_{u,a_{t_q}} < Y \leq P_{u,a_{t_{q+1}}}$ **then**
    $f(u) \leftarrow a_{t_q}$
    Set $\chi_{u,a_\iota} = 1$ for $1 \leq \iota \leq t_q$, and set $\chi_{u,a_\iota} = 0$ for $t_q < \iota \leq p+1$
procedure Shift-Right(f, v, u, a_i, b_j, Y)
  Let $b_{t_1}, b_{t_2}, \ldots, b_{t_k}$ be the neighbors of a_i in H before b_j
  Let $P_v \leftarrow \sum_{l=1}^{k} (x_{v,b_{t_l}} - x_{v,b_{t_l+1}})$, and let $P_{v,b_{t_q}} \leftarrow \sum_{l=1}^{q} (x_{v,b_{t_l}} - x_{v,b_{t_l+1}})/P_v$
  if $P_{v,b_{t_q}} < Y \leq P_{v,b_{t_{q+1}}}$ **then**
    $f(v) \leftarrow b_{t_q}$
    Set $\chi_{v,b_\iota} = 1$ for $1 \leq \iota \leq t_q$, and set $\chi_{v,b_\iota} = 0$ for $t_q < \iota \leq p+1$

Lemma 3.7. *The mapping f returned at line 7 of Algorithm 1 is a homomorphism from G to H'.*

Proof. Consider the edge $uv \in E(G)$ and suppose $f(u) = a_i$ and $f(v) = b_j$. Thus, we have $x_{u,a_{i+1}} < X \leq x_{u,a_i}$, and $x_{v,b_{j+1}} < X \leq x_{v,b_j}$. Now, by constraint (C5), we have $x_{u,a_i} \leq x_{v,b_{r(i)}}$,

Algorithm 1 Rounding the fractional values of $\mathcal{S}$

procedure ROUNDING-SHIFTING($\mathcal{S}$)
    Let $\{x_{u,a_i}\}$ and $\{x_{v,b_j}\}$ be the (fractional) values returned by solving $\mathcal{S}$
    Sample $X \in [0,1]$ uniformly at random
    For all x_{u,a_i} : if $X \le x_{u,a_i}$ set $\chi_{u,a_i} = 1$, else set $\chi_{u,a_i} = 0$
    For all x_{v,b_j} : if $X \le x_{v,b_j}$ set $\chi_{v,b_j} = 1$, else set $\chi_{v,b_j} = 0$
    Set $f(u) = a_i$ where $\chi_{u,a_i} = 1$, $\chi_{u,a_{i+1}} = 0$
    Set $f(v) = b_j$ where $\chi_{v,b_j} = 1$, $\chi_{v,b_{j+1}} = 0$
        ▷ At this point f is a homomorphism from G to H'.
    Let $F(G) = \{(u, v, f(u), f(v)) | uv \in E(G), f(u)f(v) \in E'\}$.
    Let $F \subset E'$ be the set of edges $a_i b_j$ with some $(u, v, a_i, b_j) \in F(G)$
    Choose a random variable Y with values in $[0,1]$
    while $\exists$ edge $a_i b_j \in F$ with $i + j$ is maximum **do**
        while $\exists (u, v, a_i, b_j) \in F(G)$ **do**
            if a_i does not have a neighbor after b_j and $f(u) = a_i$ **then**
                SHIFT-LEFT(f, u, v, a_i, b_j, Y)
            else if b_j does not have a neighbor after a_i and $f(v) = b_j$ **then**
                SHIFT-RIGHT(f, v, u, a_i, b_j, Y)
            Remove (u, v, a_i, b_j) from $F(G)$
        Remove $a_i b_j$ from F
        ▷ At this point f is a homomorphism from G to H.
    return f ▷ f is a homomorphism from G to H.

and hence $X \le x_{v,b_{r(i)}}$. Since $x_{v,b_{j+1}} < X$, by constraint (C3), we have $r(i) \le j$. Similarly, using the same argument for $\ell(j)$, we conclude that $\ell(j) \le i$. Therefore, a_i has a neighbor not after b_j, and b_j has a neighbor not after a_i. Now, either $a_i a_j \in E(H)$, or by the definition of E', $a_i b_j \in E'$. □

Let W denote the value of the objective function with the fractional optimum x_{u,a_i}, x_{v,b_j}, and W' denote the value of the objective function with the final values $\chi_{u,a_i}, \chi_{v,b_j}$, after the rounding and all the shifting. Also, let W^* be the minimum cost of a homomorphism from G to H. Obviously, $W \le W^* \le W'$. We now show that the expected value of W' is at most a constant times W.

Lemma 3.8. *Algorithm 1 runs in polynomial-time and it returns the homormorphism f from G to H such that for $u, v \in G$ and $a_t, b_j \in H$ we have*

$$\mathbb{P}\left[\chi_{u,a_t} = 1, \chi_{u,a_{t+1}} = 0 \text{ i.e. } f(u) = a_t\right] \le x_{u,a_t} - x_{u,a_{t+1}} \tag{1}$$

$$\mathbb{P}\left[\chi_{v,b_j} = 1, \chi_{v,b_{j+1}} = 0 \text{ i.e. } f(v) = b_j\right] \le x_{v,b_j} - x_{v,b_{j+1}} \tag{2}$$

Moreover, the expected contribution of each summand, say $c(u, a_t)(\chi_{u,a_t} - \chi_{u,a_{t+1}})$, to the expected cost of W' is at most $|V(H)|c(u, a_t)(x_{u,a_t} - x_{u,a_{t+1}})$.

Proof. Recall that after the rounding step using the random variable X, we have a partial homomorphism $f : V(G) \to V(H)$, where $f(u) = a_i$ if $x_{u,a_{i+1}} < X \leq x_{u,a_i}$, and $f(v) = b_j$ if $x_{v,b_{j+1}} < X \leq x_{v,b_j}$. By Lemma 3.7, f is a homomorphism from G to H'. We show the following claims, which help us through the rest of the proof.

Claim 3.9. *Let $uv, uw \in E(G)$. Suppose $f(u)f(v) \in E'$, and $f(u)f(w) \in E(H)$. After shifting the image of u to a_t, we have $a_t f(w) \in E(H)$.*

Proof. Let $f(u) = a_i$ and $f(v) = b_j$ and $a_i b_j \notin E(H)$, and $a_i a_l \in E(H)$ where $b_l = f(w)$. Since we have shifted the image of u in Algorithm 1, according to Observation 3.5, a_i has no neighbor after b_j. Now because $a_i b_l \in E(H)$, we have $b_l < b_j$. Since $a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_q$ is a min ordering, and $a_i b_l, a_t b_j \in E(H)$ with $t < i$, $l < j$, we conclude that $a_t b_l \in E(H)$. □

Claim 3.10. *Let $uv, uw \in E(G)$. Suppose $f(u)f(v) \in E'$. Suppose that the image of u is shifted to a_t, and $a_t f(w) \notin E(H)$, then the* Shift-Right *shifts the image of $f(w)$ to a neighbor of a_t.*

Proof. Let $a_i = f(u)$, $b_j = f(v)$. Let $b_s = f(w)$. If $a_i b_s \in E(H)$, as we argued in the Claim 3.9, $a_t b_s \in E(H)$ and we don't need to change the image of w because of u. Thus, we may assume $a_t b_s \in E'$. Now since $i + j$ is maximum, $b_s < b_j$. This would imply that $a_i b_s \in E'$, and hence, we shift the image of w according to the rules of the Algorithm 1 to a neighbor of a_i, say b_l and before b_s. Now by the min ordering property $a_t b_l \in E(H)$. □

From the proof of Claims 3.9 and 3.10 the image of each vertex u is shifted at most one. We observe that the image of vertex u is always changed to a smaller element. Moreover, at each step one element is removed from $F(G)$. Suppose $uv, uw \in E(G)$. By Claim 3.9, if $f(u)f(w)$ is in $E(H)$, then by shifting the image of $f(u)$ because of uv being mapped to E', there is no need to change the image of w. Furthermore, by claim 3.10 if by shifting the image of $f(u)$ from a_i to a_t, there is no edge between $f(w)a_t$ then w is shifted to a neighbor of a_i that is also a neighbor of a_t. These conclusions guarantee that at each step the number of elements in $F(G)$ is decreased. It is clear that for each $a_i b_j$ in F, at most $|V(G)|$ shifts are needed. Therefore, Algorithm 1 runs in polynomial-time and f is a homomorphism from G to H.

According to the rules of the Algorithm 1, vertex u is mapped to a_t in two cases. The first case is where u is mapped to a_t by rounding, and is not shifted away. In other words, we have $\chi_{u,a_t} = 1$ and $\chi_{u,a_{t+1}} = 0$ after rounding, and these values do not change by procedures Shift-Left. Hence, for this case we have:

$$\mathbb{P}[f(u) = a_t] \leq \mathbb{P}[x_{u,a_{t+1}} < X \leq x_{u,a_t}] = x_{u,a_t} - x_{u,a_{t+1}}$$

where the first inequality follows from the fact that the probability that the image of u is not changed by either of shifting procedures is at most 1. Whence, this situation occurs with probability at most $x_{u,a_t} - x_{u,a_{t+1}}$, and the expected contribution of the corresponding summand is at most $c(u, a_t)(x_{u,a_t} - x_{u,a_{t+1}})$.

Second case is where $f(u)$ is set to a_t during Shift-Left. We first calculate the contribution for a fixed i, that is, the contribution of shifting u from a fixed a_i to a_t in Shift-Left.

Note that u is first mapped to a_i, $i > t$, by rounding, and then re-mapped to a_t during procedure SHIFT-LEFT. This happens **if there exists** j and v such that uv is an edge of G, and $a_ib_j \in F \subseteq E'$ (with $i+j$ being maximum) and then the image of u is shifted to a_t ($a_t < a_i$ in the min ordering), where $a_tb_j \in E(H)$. In other words, we have $\chi_{u,a_i} = \chi_{v,b_j} = 1$ and $\chi_{u,a_{i+1}} = \chi_{v,b_{j+1}} = 0$ after rounding; and then u is shifted from a_i to a_t.

Recall that this shift occurs when a_i does not have any neighbors after b_j and Algorithm 1 calls SHIFT-LEFT. Furthermore, $a_ib_j \in F$ is chosen so that $i+j$ is maximized. We show the following claim which enables us to assume we only need to consider only one neighbor of u, namely, v in our calculation.

Claim 3.11. *, For every neighbor w of u where $X \leq x_{w,b_j}$, we must have $x_{w,b_{j+1}} < X$.*

Proof. By Observation 3.4, the ordering $a_1 < a_2 < \cdots < a_p < b_1 < b_2 < \cdots < b_p$ is a min-max ordering with respect to $E(H) \cup E'$, and by Lemma 3.7 every edge of G is mapped to an edge in $E(H) \cup E'$, after the rounding step by variable X. Suppose for some $uw \in E(G)$ we have $x_{w,b_{j+1}} \geq X$ which implies that uw is mapped to $a_ib_{j'} \in E(H) \cup E'$ with $j < j'$, this in turn contradicts our assumptions that a_i does not have any neighbor after b_j and $i+j$ is maximum.

□

By the above claim no neighbor of u is mapped to a vertex after b_j in the rounding step. By Claim 3.11 we must have $x_{w,b_{j+1}} < X$ for all w with $uw \in E(G)$. That is,

$$\alpha = \max_{w:uw \in E(G)} x_{w,b_{j+1}} < X \tag{3}$$

Define events $\mathcal{A}$ and $\mathcal{B}$ as follows:

Event $\mathcal{A}$: there exists v such that uv is an edge of G, and u is mapped to a_i and v is mapped to b_j during rounding step. That is the event $\chi_{u,a_i} = \chi_{v,b_j} = 1, \chi_{u,a_{i+1}} = \chi_{v,b_{j+1}} = 0$.

Event $\mathcal{B}$: the image of u is shifted to a_t from a_i ($t < i$). That is the event $P_{u,a_{t_j}} < Y \leq P_{u,a_{t_{j+1}}}$.

Consider event $\mathcal{A}$ and two cases where b_j has some neighbor after a_i and the case where b_j does not have a neighbor after a_i. Let C be the non-empty set of indices $C = \{t \mid t < i, a_tb_j \in E(H)\}$. In the first case, we have:

$$
\begin{aligned}
\mathbb{P}\left[\text{event } \mathcal{A} \text{ happens}\right] &= \mathbb{P}\left[\exists uw \in E(G) : \chi_{u,a_i} = \chi_{w,b_j} = 1, \chi_{u,a_{i+1}} = \chi_{w,b_{j+1}} = 0\right] && (4)\\
&= \mathbb{P}\left[\exists uw \in E(G) : \max\{x_{u,a_{i+1}}, \alpha\} < X \leq \min\{x_{u,a_i}, x_{w,b_j}\}\right] && (5)\\
&\leq \min\left\{x_{u,a_i}, \max_{w:uw\in E(G)} x_{w,b_j}\right\} - \max\left\{x_{u,a_{i+1}}, \alpha\right\} && (6)\\
&\leq x_{v,b_j} - x_{u,a_{i+1}} && (\ v = \operatorname*{argmax}_{w:uw\in E(G)} x_{w,b_j})\\
&\leq x_{v,b_j} - x_{u,a_s} && (\ a_s \text{ is the first neighbor of } b_j \text{ after } a_i\text{, and we have } x_{u,a_s} \leq x_{u,a_{i+1}})\\
&\leq \sum_{t\in C}(x_{u,a_t} - x_{u,a_{t+1}}) = P_u && (7)
\end{aligned}
$$

The last inequality is because a_i has no neighbor after b_j and it follows from constraint $(C5)$. Second for the case where b_j has no neighbor after a_i. By constraint (C8), for every v that is a neighbor of u we have:

$$
x_{v,b_j} - x_{v,b_{j+1}} \leq \sum_{t\in C} x_{u,a_t} - x_{u,a_{t+1}} = P_u \tag{8}
$$

We therefore obtain:

$$
\begin{aligned}
\mathbb{P}\left[\text{event } \mathcal{A} \text{ happens}\right] &= \mathbb{P}\left[\exists uw \in E(G) : \chi_{u,a_i} = \chi_{w,b_j} = 1, \chi_{u,a_{i+1}} = \chi_{w,b_{j+1}} = 0\right] && (9)\\
&= \mathbb{P}\left[\exists uw \in E(G) : \max\{x_{u,a_{i+1}}, \alpha\} < X \leq \min\{x_{u,a_i}, x_{w,b_j}\}\right] && (10)\\
&\leq \min\left\{x_{u,a_i}, \max_{w:uw\in E(G)} x_{w,b_j}\right\} - \max\left\{x_{u,a_{i+1}}, \alpha\right\} && (11)\\
&\leq x_{v,b_j} - \alpha && (\ v = \operatorname*{argmax}_{w:uw\in E(G)} x_{w,b_j})\\
&\leq x_{v,b_{j+1}} + P_u - \alpha && (\text{by } (8))\\
&\leq x_{v,b_{j+1}} + P_u - x_{v,b_{j+1}} && (\text{by } (3))\\
&= P_u && (12)
\end{aligned}
$$

Having uv mapped to a_ib_j in the rounding step, we shift u to a_t with probability $P_{u,t} = (x_{u,a_t} - x_{u,a_{t+1}})/P_u$. That is $\mathbb{P}[\mathcal{B} \mid \mathcal{A}] = P_{u,t}$. Note that the upper bound $\mathbb{P}[\mathcal{A}] \leq P_u$ is independent from the choice of v and b_j. Moreover, recall that random variables X and Y are independent. Therefore, for a fixed a_i, the probability that u is shifted from a_i to a_t is at most

$$
\mathbb{P}[\mathcal{B} \mid \mathcal{A}] \cdot \mathbb{P}[\mathcal{A}] \leq \ ((x_{u,a_t} - x_{u,a_{t+1}})/P_u) \cdot P_u = x_{u,a_t} - x_{u,a_{t+1}}
$$

Thus, the expected contribution for a fixed a_i (with its b_j and v) is also at most $c(u,a_t)(x_{u,a_t} - x_{u,a_{t+1}})$. Notice that there are at most $|V(H)|-1$ of such a_i's, thus the expected contribution of $c(u,a_t)$ to the expected value of W' is at most $|V(H)|c(u,a_t)(x_{u,a_t} - x_{u,a_{t+1}})$.

□

Theorem 3.12. *Algorithm 1 returns a homomorphism with expected cost at most $|V(H)|$ times optimal solution. The algorithm can be derandomized to obtain a deterministic $|V(H)|$-approximation algorithm.*

Proof. By Lemma 3.8 and linearity of expectation, for the expected value of W' we have

$$\begin{aligned}\mathbb{E}[W'] &= \mathbb{E}\left[\sum_{u,i} c(u,a_i)(\chi_{u,a_i} - \chi_{u,a_{i+1}}) + \sum_{v,j} c(v,b_j)(\chi_{v,b_j} - \chi_{v,b_{j+1}})\right] \\ &= \sum_{u,i} c(u,a_i)\mathbb{E}[\chi_{u,a_i} - \chi_{u,a_{i+1}}] + \sum_{v,j} c(v,b_j)\mathbb{E}[\chi_{v,b_j} - \chi_{v,b_{j+1}}] \\ &\leq |V(H)|(\sum_{u,i} c(u,a_i)(x_{u,a_i} - x_{u,a_{i+1}}) + \sum_{v,j} c(v,b_j)(\chi_{v,b_j} - \chi_{v,b_{j+1}})) \\ &\leq |V(H)|W \leq |V(H)|W^*.\end{aligned}$$

Thus Algorithm 1 outputs a homomorphism whose expected cost is at most $|V(H)|$ times the minimum cost. It can be transformed to a deterministic algorithm as follows. There are only polynomially many values x_{u,a_i}, x_{v,b_j} (at most $|V(G)| \cdot |V(H)|$). When X lies anywhere between two such consecutive values, all computations will remain the same. Similarly, there are only polynomially many values of the partial sums $\sum_{p=1}^{q} P_{u,t_p}$, and when Y lies anywhere between two consecutive values, all the computations remain the same. Moreover, for any given X and Y, the rounding and shifting algorithms can be performed in polynomial time. Thus, we can derandomize the algorithm by trying all the possible values for X and Y and simply choose the pair that gives us the minimum homomorphism cost. Since the expected value is at most $|V(H)|$ times the minimum cost, this bound also applies to this best solution. □

4 A dichotomy for graphs

Feder *et al.,* [8] showed that LHOM(H) is polynomial-time solvable if and only if H is a *bi-arc* graph. Bi-arc graphs are defined as follows.

Let C be a circle with two specified points p and q on C. A bi-arc is an ordered pair of arcs (N, S) on C such that N contains p but not q, and S contains q but not p. A graph H is a bi-arc graph if there is a family of bi-arcs $\{(N_x, S_x) : x \in V(H)\}$ such that, for any $x, y \in V(H)$, not necessarily distinct, the following hold:

- if x and y are adjacent, then neither N_x intersects S_y nor N_y intersects S_x;
- if x and y are not adjacent, then N_x intersects S_y and N_y intersects S_x.

We shall refer to $\{(N_x, S_x) : x \in V(H)\}$ as a bi-arc representation of H. Note that a bi-arc representation cannot contain bi-arcs $(N, S), (N', S')$ such that N intersects S' but S does not intersect N' and vice versa. Furthermore, by the above definition a vertex may have a self loop.

Theorem 4.1 ([4, 8]). *A graph admits a conservative majority polymorphism if and only if it is a bi-arc graph.*

Definition 4.2 (H^*). *Let $H = (V, E)$ be a graph. Let H^* be a bipartite graph with partite sets V, V' where V' is a copy of V. Two vertices $u \in V$, and $v' \in V'$ of H^* are adjacent in H^* if and only if uv is an edge of H.*

Lemma 4.3. *Let H^* be the bipartite graph constructed from a bi-arc graph H, according to Definition 4.2. Then the following hold.*

- *H^* is a co-circular arc graph.*
- *H^* admits a min-ordering.*

Proof. It is easy to see that H^* is a co-circular arc graph. From a bi-arc representation $\{(N_i, S_i) : i \in V(H)\}$ of H, we obtain a co-circular arc representation of H^* by choosing, for $i \in H$, the arc N_i for vertex $i \in H^*$ and the arc S_i for vertex $i' \in H^*$. A bipartite graph admits a min-ordering if and only if it is co-circular arc graph [16]. H^* is a co-circular arc graph, and hence, it admits a min-ordering. □

Construction of H^* and choosing a min ordering Let H be a bi-arc graph, with vertex set I, and let H^* be the bipartite graph constructed from H having vertices (I, I') according to Definition 4.2. Let $a_1, a_2, \ldots, a_p$ be an ordering of the vertices in I and $b_1, b_2, \ldots, b_p$ be an ordering of the vertices of I'. Note that each a_i has a copy $b_{\pi(i)}$ in $\{b_1, b_2, \ldots, b_n\}$ where π is a permutation on $\{1, 2, 3, \ldots, p\}$. By Lemma 4.3, let us assume $a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_p$ is a min-ordering for H^*. For every a_i, let $r(i)$ be the smallest subscript such that $a_i b_{r(i)}$ is an edge of H^* and for every b_j, let $\ell(j)$ be the smallest subscript such that $a_{\ell(j)} b_j$ is an edge of H^*.

Let G be the input graph with vertex set V and let c be a given cost function. Construct G^* from G with vertex set $V \cup V'$ as in Definition 4.2. Now construct an instance of the MinHOM(H^*) for the input graph G^* and set $c(v', b_{\pi(i)}) = c(v, a_i)$ for $v \in V$, $v' \in V'$.

Lemma 4.4. *There exists a homomorphism $f : G \to H$ with cost $\mathfrak{C}$ if and only if there exists homomorphism $f^* : G^* \to H^*$ with cost $2\mathfrak{C}$ such that, if $f^*(v) = a_i$ then $f^*(v') = b_j$ with $j = \pi(i)$.*

Introducing the lists Let $G = (V, E(G))$ be our input bipartite graph. We assume G is connected.

To each vertex $u \in V$, we associate a list $L(v)$ that initially contains $V(H)$. Think of $L(u)$ as the set of possible images for u in a homomorphism from G to H.

Apply the *arc consistency* procedure as follows. Take an arbitrary edge $xy \in E(G)$ and let $a \in L(x)$. If there is no neighbor of a in $L(y)$ then remove a from $L(x)$. Repeat this until a list becomes empty or no more changes can be made. Note that if we end up with an empty list after arc consistency, then there is no homomorphism of G to H. After the arc consistency check, we perform the pair consistency check.

$$\texttt{Minimize} \sum_{v,i} c(v,a_i)(x_{v,a_i} - x_{v,a_{i+1}}) + \sum_{v',j} c(v',b_i)(x_{v',b_j} - x_{v',b_{j+1}})$$

Subject to:

$x_{v,a_i}, x_{v',b_{\pi(i)}} \geq 0$	$\forall v, v' \in G^*, a_i, b_{\pi(i)} \in H^*$	(CM1)
$x_{v,a_1} = x_{v',b_1} = 1$		(CM2)
$x_{v,a_{p+1}} = x_{v',b_{p+1}} = 0$		(CM3)
$x_{v,a_{i+1}} \leq x_{v,a_i}$ and $x_{v',b_{j+1}} \leq x_{v',b_j}$	$\forall v, v' \in G^*, a_i, b_j \in H^*$	(CM4)
$x_{v,a_{i+1}} = x_{v,a_i}$ and $x_{v',b_{\pi(i)+1}} = x_{v',b_{\pi(i)}}$	$\forall v \in V(G^*), a_i \in V(H)$ if $a_i \notin L(v)$	(CM5)
$x_{u,a_i} \leq x_{v',b_{r(i)}}$ and $x_{v',b_i} \leq x_{u,a_{l(i)}}$	$\forall uv \in E(G^*)$	(CM6)
$x_{u,a_i} - x_{u,a_{i+1}} = x_{u',b_{\pi(i)}} - x_{u',b_{\pi(i)+1}}$	$\forall u, u' \in G^*, \forall a_i, b_{\pi(i)} \in H^*$	(CM7)
$x_{v',b_j} \leq x_{u,a_s} + \sum_{\substack{t<i \\ a_t b_j \in E \\ a_t \in L(u)}} (x_{u,a_t} - x_{u,a_{t+1}})$	$\forall uv' \in E(G^*), a_i b_j \in E'$, and a_s is the first neighbor of b_j after a_i in $L(u)$	(CM8)
$x_{u,a_i} \leq x_{v',b_s} + \sum_{\substack{t<j \\ a_i b_t \in E \\ a_t \in L(v')}} (x_{v',b_t} - x_{v',b_{t+1}})$	$\forall uv' \in E(G^*), a_i b_j \in E'$, and b_s is the first neighbor of a_i after b_j in $L(v')$	(CM9)
$x_{u,a_i} - x_{u,a_{i+1}} \leq \sum_{\substack{j: \\ (a_i,a_j) \in L(u,v)}} (x_{v,a_j} - x_{v,a_{j+1}})$	$\forall u, v \in G^*$	(CM10)

Table 2: Linear program $\mathcal{S}^*$

After the arc consistency process, the pair lists L lists are initialized by setting $L(x,y) = \{(a,b) \mid a \in L(x), b \in L(y)\}$ for every $x, y \in G$. Now for every $x, y \in G$ and every $(a,b) \in L(x,y)$, if there exists z such that for every $c \in L(z)$ either $(a,c) \notin L(x,z)$ or $(b,c) \notin L(y,z)$ then we remove (a,b) from $L(x,y)$. We continue this process until no list can be modified. If for some $a \in L(x)$, there is some $y \in D$ so that a does not appear as the first component of any pair in $L(x,y)$, then a is removed from $L(x)$. In the end, if there is any empty list, then clearly there is no homomorphism from D to H. Therefore, in the rest of the paper, we assume that all lists are non-empty. We extend the lists to G^* where $L(u)$ contains the element a_i if and only if $L(u')$ contains $b_{\pi(i)}$.

Consider the system of linear equations $\mathcal{S}^*$. For every vertex $v \in V$ from $V(G^*) = V \cup V'$ and every vertex $a_i \in I$ from $V(H^*) = I \cup I'$ define a variable x_{v,a_i}. For every vertex $v' \in V'$ from $V(G^*)$ and every vertex $b_i \in I'$ from $V(H^*)$ define a variable x_{v',b_i}. We also define the variables $x_{v,a_{p+1}}, x_{v',b_{p+1}}$ for every $v \in V$ whose value is set to zero. Now the goal is to solve the following linear program $\mathcal{S}^*$ depicted in Tabble 2:

Let E' denote the set of all pairs (a_i, b_j) such that $a_i b_j$ is not an edge of H^*, but there is an edge $a_i b_{j'}$ of H^* with $j' < j$ and an edge $a_{i'} b_j$ of H^* with $i' < i$. Define H'^* to be bipartite graph with vertex set $V(H^*)$ and edge set $E(H^*) \cup E'$. Note that $E(H^*)$ and E' are disjoint sets.

Lemma 4.5. *There is a one-to-one correspondence between homomorphisms from G to H*

and integer solutions of $\mathcal{S}^*$.

Proof. For a homomorphism $f : G \to H$, if $f(v) = a_t$ we set $x_{v,a_i} = 1$ for all $i \leq t$, otherwise, we set $x_{v,a_i} = 0$, we also set $x_{v',b_j} = 1$ for all $j \leq \pi(t)$ else set $x_{v',b_j} = 0$. We set $x_{v,a_1} = 1$, $x_{v',a_1} = 1$ and $x_{v,a_{p+1}} = x_{v',b_{p+1}} = 0$ for all $v, v' \in V(G^*)$. Now all the variables are non-negative and we have $x_{v,a_{i+1}} \leq x_{v,a_i}$ and $x_{v',b_{j+1}} \leq x_{v',b_j}$. Observe that by this assignment, the constraint (CM1)-(CM7) are satisfied.

Now for all u and v in D with $f(u) = a_i$ and $f(v) = a_j$ we have $x_{u,a_i} - x_{u,a_{i+1}} = x_{v,a_j} - x_{v,a_{j+1}} = 1$. Moreover, since f is a homomorphism, we have $(a_i, a_j) \in L(u, v)$, and hence, constraint (CM10) is also satisfied.

We show that constraint (CM8) holds. For, contradiction, assume that the inequality in (CM8) fails. This means that for some edge uv' of G^* and some edge $a_ib_j \in E'$ (the extra edges added into to make the ordering of H^*, a min-max ordering, we have $x_{v',b_j} = 1$, $x_{u,a_s} = 0$, and the sum of $x_{u,a_t} - x_{u,a_{t+1}}$ (over all $t < i$ such that a_t is a neighbor of b_j) is zero. The latter two facts imply that $f(u) = a_i$. Since b_j has a neighbor after a_i, Observation 2 tells us that a_i has no neighbor after b_j, whence $f(v') = b_j$ and thus $a_ib_j \in E(H^*)$, a contradiction the fact that $a_ib_j \in E'$. By a similar argument (CM9) is also satisfied.

Conversely, from an integer solution for $\mathcal{S}^*$, we define a homomorphism f from D to H as follows. For every $u \in D$, set $f(u) = a_i$ when i is the largest subscript with $x_{u,a_i} = 1$. Let uv be an edge of G and assume that $f(u) = a_i$, $f(v) = a_j$. Note that $x_{u,a_i} - x_{u,a_{i+1}} = x_{v,a_j} - x_{v,a_{j+1}} = 1$ and for all other $s \neq j$ we have $x_{v,a_s} - x_{v,a_{s+1}} = 0$. Since constraint (CM9) is satisfied,

$$1 = x_{u,a_i} - x_{u,a_{i+1}} \leq \sum_{(a_i,a_s) \in L(u,v)} (x_{v,a_s} - x_{v,a_{s+1}})$$

where j is the only index with $x_{v,a_j} - x_{v,a_{j+1}} \neq 0$. Therefore, $(a_i, a_j) \in L(u, v)$ and $a_ia_j \in E(H)$.

□

Theorem 4.6. *Algorithm 3, given an optimal solution for the linear program $\mathcal{S}^*$, produces a homomorphism from G to H. Furthermore, the expected cost of the homomorphism returned by this algorithm is at most $2|V(H)| \cdot OPT$.*

Proof. In Algorithm 3, lines 5 and 6, for every variable x_{u,a_i}, $u \in V(G^*)$, set $\chi_{u,a_i} = 1$ if $X \leq x_{u,a_i}$ else $\chi_{u,a_i} = 0$. Similarly, for every x_{v',b_j}, $v' \in V(G^*)$, set $\chi_{v',b_j} = 1$ if $X \leq x_{v',b_j}$ else $\chi_{v',b_i} = 0$. Let $f(u) = a_i$ where i is the largest subscript with $\chi_{u,a_i} = 1$, and let $f(v') = b_j$ where j is the largest subscript with $\chi_{v',b_j} = 1$. Notice that similar to the argument as in Claim 3.7, the mapping f produced in Line 6 of Algorithm 3, maps the edges of G^* to $E(H^*) \cup E'$. The algorithm has two stages after rounding the fractional solution using the random variable X.

Stage 1. Modifying f so that it becomes a homomorphism from G^* to H^*. Choose a random variable $Y \in [0, 1]$. Let F be the subset of edges in E' for which there exists an edge $uv' \in E(G^*)$ where uv' is mapped to that edge. Let $a_ib_j \in F$ where $i + j$ is maximum

Algorithm 3 Approximation MinHOM(H) for graphs

procedure APPROX–GRAPH-MINHOM(H)
  Construct H^*, G^* from H, G respectively, as in Definition 4.2
  Let x_{u,a_i}, u'_j s be the (fractional) values returned after solving LP $\widehat{\mathcal{S}^*}$.
  Sample X uniformly from $[0, 1]$
  For all x_{u,a_i}s: if $X \leq x_{u,a_i}$ let $\chi_{u,a_i} = 1$, else let $\chi_{u,a_i} = 0$, and $\chi_{v',b_j} = 1$ if $X \leq x_{v',b_j}$ else $\chi_{v',b_j} = 0$
  Let $f(u) = a_i$ where i is the largest subscript with $\chi_{u,a_i} = 1$, and let $f(v') = b_j$ where j is the largest subscript with $\chi_{v',b_j} = 1$,
    ▷ f is a homomorphism from G^* to $(H^*)'$
  Sample Y uniformly from $[0, 1]$
  Let $F(G^*) = \{(u, v', f(u), f(v')) \mid uv' \in E(G^*), f(u)f(v') \in E'\}$
  $F \subset E'$ be the set of edges a_ib_j with some $(u, v, a_i, b_j) \in F(G^*)$.
  while $\exists$ edge $a_ib_j \in F$ with $i + j$ is maximum **do**
    while $\exists(u, v', a_i, b_j) \in F(G^*)$ **do**
      if a_i does not have a neighbor after b_j and $f(u) = a_i$ **then**
        SHIFT-LEFT(f, u, v', a_i, b_j, Y)
      else if b_j does not have a neighbor after a_i and $f(v') = b_j$ **then**
        SHIFT-RIGHT(f, v', u, a_i, b_j, Y)
      Remove (u, v', a_i, b_j) from $F(G^*)$
    Remove a_ib_j from F
    ▷ At this point f is a homomorphism from G^* to H^*.
  Let f be the homomorphism from G^* to H^* returned in the previous step
  f =SHIFT(f)
  return f ▷ f is a homomorphism from G to H

and for some $uv' \in E(G^*)$, $f(u) = a_i$ and $f(v) = b_j$. Similar to Observation 2, either b_j has no neighbor after a_i or a_i has no neighbor after b_j. Suppose the former is the case.

Random variable $Y \in [0, 1]$ is used as guide to shift the image of v' from b_j to some b_t where $a_ib_t \in E(H^*)$, and b_t appears before b_j in the min-ordering of H^*. Consider the set of such b_ts (by definition of the min-ordering of H^*, this set is non-empty), and suppose it consists of b_t with subscripts t ordered as $t_1 < t_2 < \dots t_k$. Let $P_{v',t} = \frac{x_{v',b_t} - x_{v',b_{t+1}}}{P_{v'}}$ with $P_{v'} = \sum_{a_ib_t \in E(H^*),\ t<j} (x_{v',b_t} - x_{v',b_{t+1}})$. Select the vertex b_{t_q} if $\sum_{p=1}^{q} P_{v',t_p} < Y \leq \sum_{p=1}^{q+1} P_{v',t_p}$. Thus, b_t is selected with probability $P_{v',t}$, which is proportional to the difference of fractional values $x_{v',b_t} - x_{v',b_{t+1}}$.

The proof of the following Claim is similar to Claim 3.7.

Claim 4.7. *Let w be a neighbor of v', where $f(w) = a_s$ and $a_sb_j \in E(H^*) \cup E'$. Then $f(w)b_t \in E(H^*) \cup E'$.*

Proof. Proof is almost identical to the proof of Claim 3.7. ☐

Algorithm 4 Procedures Shift-Left and Shift-Right

procedure Shift-Left(f, u, v', a_i, b_j, Y)
  Let $a_{t_1}, a_{t_2}, \ldots, a_{t_k}$ be the neighbors of b_j in $L(u)$ and before a_i
  Let $P_u \leftarrow \sum_{l=1}^{k}(x_{u,a_{t_l}} - x_{u,a_{t_l+1}})$, and let $P_{u,t_j} \leftarrow \sum_{l=1}^{j}(x_{u,a_{t_l}} - x_{u,a_{t_l+1}})/P_u$
  if $P_{u,t_j} < Y \leq P_{u,t_{j+1}}$ **then**
    $f(u) \leftarrow a_{t_j}$
procedure Shift-Right(f, v', u, a_i, b_j, Y)
  Let $b_{t_1}, b_{t_2}, \ldots, b_{t_k}$ be the neighbors of a_i in $L(v')$ and before b_j
  Let $P_{v'} \leftarrow \sum_{l=1}^{k}(x_{v',b_{t_l}} - x_{v',b_{t_l+1}})$, and let $P_{v',t_j} \leftarrow \sum_{l=1}^{j}(x_{v',b_{t_l}} - x_{v',b_{t_l+1}})/P_{v'}$
  if $P_{v',t_j} < Y \leq P_{v',t_{j+1}}$ **then**
    $f(v') \leftarrow b_{t_j}$

Note that as long as F is not empty, we repeat the shifting procedure. By Claim 4.7 after each shift the resulting f is a homomorphism from G^* to the graph induced by edges $E(H^*) \cup E'$. Once, there is no edges of G^* whose imgae under f is mapped to E'; i.e. F is empty, f is a homomorphism from G^* to H^*.

Algorithm 5 The shifting procedure for unstable vertices (Stage 2)

procedure Shift(f)
  while there are unstable vertices **do**
    Let u be a vertex with $f(u) = a_i$ and $f(u') \neq b_{\pi(i)}$ where i is maximum.
    Let Q be a Queue. $Q.enqueue(u')$
    while Q is not empty **do**
      $x \leftarrow Q.dequeue()$
      if $x = v'$ **then**
        $f(v') \leftarrow b_{\pi(i)}$ where $f(v) = a_i$.
        for $wv' \in E(D)$ with $a_\ell = f(w)$ and $f(w') \neq b_{\pi(\ell)}$ **do**
          $Q.enqueue(w)$
      else if $x = v$ **then**
        $f(v) \leftarrow a_i$ where $f(v') = b_{\pi(i)}$.
        for $vw' \in E(D)$ with $a_\ell = f(w)$ and $f(w') \neq b_{\pi(\ell)}$ **do**
          $Q.enqueue(w')$
  return f ▷ f is a homomorphism from G to H

Stage 2. Making the assignment consistent with respect to both orderings: We say a vertex $u \in V$ is *unstable* if $f(u) = a_i$, $f(u') = b_q$ where $q \neq \pi(i)$. Now we start a BFS in $V(G^*)$ and continue as long as there exists an unstable vertex. At each step, we start from the greatest subscripts i for which there exists an unstable u with $f(u) = a_i$. During the BFS, one of the following is performed:

1. shift the image of u' from b_q to $b_{\pi(i)}$.
2. shift the image of u from a_i to $a_{\pi^{-1}(q)}$.

As a consequence of the above actions, we would have the following cases:

Case 1: We change the image of u' from b_q to $b_{\pi(i)}$ (with $f(u) = a_i$), and there exists some $v' \in V'$ such that $uv' \in E(G^*)$ with $f(v) = a_j$ and $f(v') = b_{\pi(j)}$.

We note that $a_i b_{\pi(j)}$ is an edge because uv' is an edge, and hence, $a_j b_{\pi(i)}$ is an edge of H^*. This would mean there is no need to shift the image of v from a_j to something else (see the Figure 1a).

Case 2: We change the image of u' from b_q to $b_{\pi(i)}$ (with $f(u) = a_i$), and there exists some edge vu' of H^* with $f(v) = a_j$ and $f(v') = b_\ell$ with $\ell \neq \pi(j)$.

Such vertex v is added into the queue, and once we retrieve v from the queue we do the following: changing the image of v from a_j to $a_{\pi^{-1}(\ell)}$ (see the Figure 1b).

Note that $a_i b_\ell \in E(H^*)$ because vu' is an edge of G^*, and hence $a_{\pi^{-1}(\ell)} b_{\pi(i)}$ is an edge of H^*.

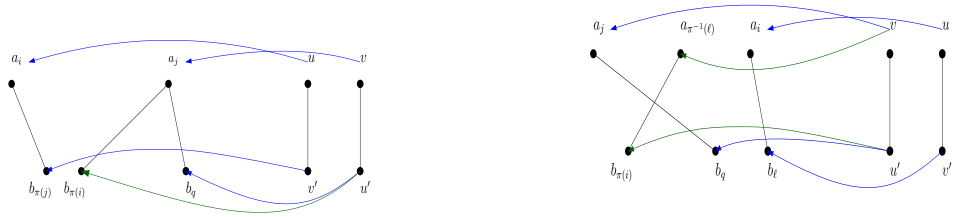
Case 3: We change the image of v from a_j to some $a_{\pi^{-1}(\ell)}$ (with $f(v') = b_{\pi(\ell)}$) and there exists some vw' such that $f(w) = a_t$ and $f(w') = b_{\pi(t)}$. We note that $a_t b_\ell \in E(H^*)$ because $v'w$ is an edge, and hence, $a_{\pi^{-1}(\ell)} b_r$ is an edge of H^*. This would mean there is no need to shift the image of w' to something else.

Case 4: We change the image of v from a_j to some $a_{\pi^{-1}(\ell)}$ (with $f(v') = b_\ell$). Let r be a greatest subscript such that there exists some vw' where $f(w) = a_t$ and $f(w') = b_r$ with $r \neq \pi(t)$, $t < i$. Such vertex w' is added into the queue, and once we retrieve w' from the queue we do the following: changing the image of w' from b_r to $b_{\pi^{-1}(t)}$.

Note that $a_t b_\ell \in E(H^*)$ because wv' is also an edge of G^*. Hence, $a_{\pi^{-1}(\ell)} b_{\pi^{-1}(t)}$ is an edge of H^*.

When Case 2 occurs, we continue the shifting. This would mean we may need to shift the image of some neighbor w' of v accordingly. We continue the BFS from v, and modify the images of neighbors of v, say w', to be consistent with new image of v. This means we encounter either Case 3 or Case 4. Suppose $f(w') = b_t$ or $f(w') = b_{\pi(t)}$ Then there is no need to change the image of w'. Otherwise, we change the image of w' from b_t to b_j where $a_{\pi^{-1}(\ell)} b_j$ is an edge of H^* and we need to consider Cases 3,4 for the current vertex w. When we are in Case 4, then consider Cases 1,2 and proceed accordingly.

During the BFS, the image of a stable vertex remains unchanged, as specified in Cases 1 and 3. This holds true not only for pre-existing stable vertices but also for vertices that become stable as the algorithm progresses. Furthermore, as the algorithm progresses, the number of unstable vertices consistently decreases. Consequently, the entire process terminates after, at most $O(|V(G)|)$ iterations.



(a) Case 1

(b) Case 2

Figure 1: Illustrating the shifting process in Stage 2 of the algorithm.

Estimating the ratio. Vertex v (v', resp.) is mapped to a_t (b_t, resp.) in three situations. The first scenario is where v is mapped to a_t by rounding (according to random variable X in Stage 1) and is not shifted away. In other words, we have $\chi_{v,a_t} = 1$ and $\chi_{v,a_{t+1}} = 0$ (i.e. $x_{v,a_{t+1}} \leq X < x_{v,a_t}$) and these values do not change by the shifting procedure. Hence, for this case we have: $\mathbb{P}[f(v) = a_t] = \mathbb{P}[x_{v,a_{t+1}} < X \leq x_{v,a_t}] \leq x_{v,a_t} - x_{v,a_{t+1}}$. Whence this situation occurs with probability at most $x_{v,a_t} - x_{v,a_{t+1}}$, and the expected contribution is at most $c(v, a_t)(x_{v,a_t} - x_{v,a_{t+1}})$.
The second scenario is where $f(v)$ is set to a_t according to the random variable Y in Stage 1. In this case v is first mapped to $a_j, j > t$, by rounding according to variable X and then re-mapped to a_t during the shifting according to variable Y. Similar to the argument in Lemma 3.8 this situation occurs with probability at most $x_{v,a_t} - x_{v,a_{t+1}}$. Therefore, the expected contribution of $x_{v,a_t} - x_{v,a_{t+1}}$ to the objective function is at most $|V(H)|c(v, a_t)(x_{v,a_t} - x_{v,a_{t+1}})$. The third scenario is when the image of v is shifted from some a_j to a_t in the second Stage of the shifting. More precisely, when one of the actions 1,2 occurs. This happens because the image of v' has been shifted to $b_{\pi(t)}$ in Stage 2 according to variables X or Y (i.e. BFS). As we argued, in the previous scenarios in Stage 1, the overall expected contribution of $c(v', b_{\pi(t)})$ into the objective function is $|V(H)|c(v, a_t)(x_{v',b_{\pi(t)}} - x_{v',b_{\pi(t)+1}})$. In Stage 2, we shift the image of v to a_t because v is unstable and the image of v' is $b_{\pi(t)}$. In Stage 1, the expected contribution of $c(v, a_t)$ into the objective function is $|V(H)|c(v, a_t)(x_{v,a_t} - x_{v,a_{t+1}})$. Since $x_{v,a_t} - x_{v,a_{t+1}} = x_{v',b_{\pi(t)}} - x_{v',b_{\pi(t)+1}}$, the overall expected value of shifting v to a_t is $2|V(H)|c(v, a_t)(x_{v,a_t} - x_{v,a_{t+1}})$. □

We remark that, as in the proof of Theorem 3.12, the above algorithm can be de-randomized. By Lemma 4.3 and Theorem 4.6 we obtain the following classification theorem.

Theorem 4.8. *If H admits a conservative majority polymorphism, then MinHOM(H) has a (deterministic) $2|V(H)|$-approximation algorithm, otherwise, MinHOM(H) is inapproximable unless P=NP.*

5 Inapproximability of H-coloring for graphs

We say an optimization problem $\mathcal{P}$ is α-*approx-hard*, $\alpha > 0$, if it is NP-hard to find an α-approximation for $\mathcal{P}$. Note that if $\mathcal{P}$ is a maximization problem then $\alpha \leq 1$, and if it a minimization problem then $\alpha \geq 1$.

We also use another notion of inapproximability under the UNIQUE GAME CONJECTURE [24], UGC for short. We say an optimization problem $\mathcal{P}$ is *α-UG-hard* if it is UG-hard to approximate $\mathcal{P}$ within factor α. See [2] for further details.

A nice property of the MinHOM problem is that the hardness results for approximation are "carried over" by induced sub-graphs. This means if MinHOM(H) is α-approx-hard or it is α-UG-hard, then the same holds for any graph which has H as its induced sub-graph. Informally speaking, such a property holds since the cost functions in the MinHOM problem are part of inputs, hence, modifying cost functions gives rise to hardness results for every graph H' which has H as its induced graph. This is proved formally as follows.

Lemma 5.1. *[Hardness of approximation for sub-graph] Let H be an induced sub-graph of graph H'. If MinHOM(H) is α-approx-hard [α-UG-hard], then MinHOM(H') is α-approx-hard [α-UG-hard].*

Proof. Let G, H together with cost function $c : G \times H \to \mathbb{Q}_{\geq 0}$ be an instance of MinHOM(H). Construct an instance of MinHOM(H') with graphs G, H' and cost function $c' : G \times H' \to \mathbb{Q}_{\geq 0}$ where $c'(u,i) = c(u,i)$ for every $u \in G$ and $i \in H$, otherwise, for every $u \in G$ and $i \in H' \setminus H$, $c'(u,i) = W$ where W is a number greater than $(1 + \max\{c(u,i) \mid u \in G, i \in H\})|G|)$. Notice that the cost of any homomorphism from G to H is strictly less than W.

Notice that $f'^* : V(G) \to V(H')$, the minimum cost homomorphism from G to H', does not map any of the vertices of G to any vertex in $H' \setminus H$ due to the way we have defined c'. Therefore, f'^* is also the minimum cost homomorphism for H. Now it is straightforward to see that if an algorithm approximates $f^* : V(G) \to V(H)$, the minimum cost homomorphism from G to H within a factor α, it is, in fact, computing an α-approximation of f'^*. □

5.1 Hardness of approximation for graphs

In this subsection we prove that MinHOM for graphs does not admit any PTAS and in a sense a cosntant factor approximation is the best one can achieve. We start with the following theorems about the complexity of MinHOM(H) for graph H.

Theorem 5.2. *[11] Let H be a bipartite graph. Then MinHOM(H) is polynomial-time solvable if and only if H admits a min-max ordering (i.e., does not contain an induced cycle of length at least six, or a bipartite claw, or a bipartite net, or a bipartite tent, see Figure 2).*

Theorem 5.3. *[11] Let H be graph with at least one self-loop vertex. Then MinHOM(H) is polynomial-time solvable if and only if H is reflexive (every vertex has a self-loop) and admits a min-max ordering (i.e., does not contain an induced cycle of length at least four, or a claw, or a net, or a tent, see Figure 3).*

The obstruction to min-max ordering for graphs are invertible pairs [20]. We say two vertices a and b of graph(bipartite graph) H is an invertible pair if there exist two walks P from a to b and Q from b to a of the same length such that when $a_i a_{i+1}$, $b_i b_{i+1}$ are the

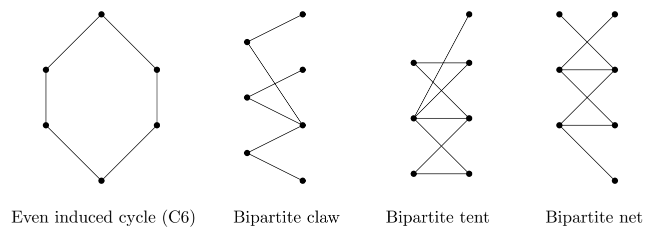


Figure 2: Obstruction to min-max ordering in bipartite graphs, and making MinHOM(H) NP-complete.

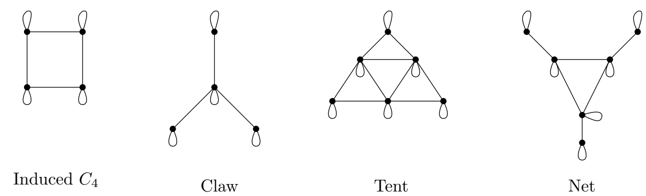


Figure 3: Obstruction to min-max ordering in reflexive graphs, and making MinHOM(H) NP-complete.

i-th edge of P and Q then at least one of the a_ib_{i+1}, b_ia_{i+1} is not an edge of H. We use the existence of these obstruction in our gap preserving approximation reduction.

Before going to the main result, recall the following lemma that establishes the relationship between non-polynomial cases of the LHOM and the approximation of MinHOM.

Lemma 5.4. *[16] If LHOM(H) is not polynomial-time solvable then MinHOM(H) does not have any approximation.*

Now, we are ready to obtain hardness of approximation for MinHOM(H) when H is a graph.

Theorem 5.5. *Let H be a graph where MinHOM(H) is NP-complete. Then MinHOM(H) is at least* 1.128*-approx-hard (under $P \neq NP$ assumption), and* 1.242*-UG-hard.*

Proof. We consider two cases, where H is irreflexive (no vertex has a self-loop) and the case where H has a vertex with self-loop.

H **is irreflexive:** Without loss of generality, we can assume H is bipartite, as otherwise, HOM(H) is **NP**-complete (due to [17]). Hence, LHOM(H) is **NP**-complete, and by Lemma 5.4, MinHOM(H) does not have any approximation. By this argument and by Lemma 5.1 (hardness of approximation for sub-graph), if a sub-graph of H is not bipartite, again MinHOM(H) does not admit any approximation. Therefore, we continue by assuming that H is bipartite. Moreover, when bipartite graph H contains an induced even cycle of length at least 6, LHOM(H) is **NP**-complete due to [7], and hence, by Lemma 5.4 MinHOM(H)

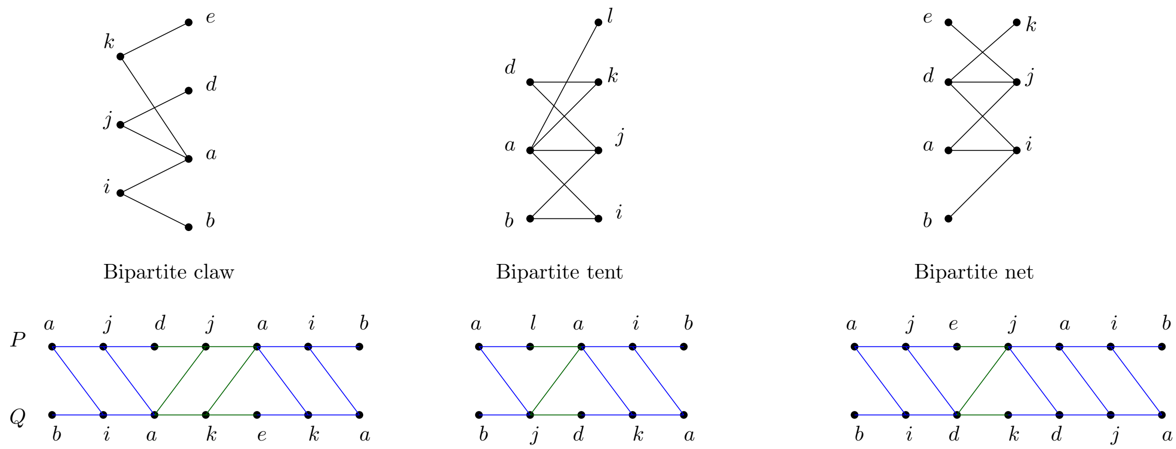


Figure 4: Invertible pair for bipartite claw, tent, and net.

admits no approximation. By Theorem 5.2 and Lemma 5.1, it remains to consider the cases where H is either bipartite claw, bipartite tent, or bipartite net.

We start with bipartite claw first. Let H be a bipartite claw with the vertex set $\{a, b, d, e, i, j, k\}$ and the edge set with edge set $\{bi, ai, aj, ak, ke, dj\}$ (as depicted in Figure 4). It was shown in [25] that it is **NP**-hard to approximate the Vertex Cover within factor better than $\sqrt{2} - \epsilon$. Vertex Cover is also $(2 - \epsilon)$-UG-hard by [26]. Let G be any of the graphs described in [5, 25], with $V(G) = \{x_1, x_2, \ldots, x_n\}$. This graph has a relatively large vertex cover.

Construction of the bipartite graph G': We construct the bipartite graph G' as follows. The vertex set of G' consists of three disjoint copies V_1, V_2, V_3 of $V(G)$ together with set U. Let $V_1 = \{u_1, u_2, \ldots, u_n\}$, $V_2 = \{v_1, v_2, \ldots, v_n\}$, and $V_3 = \{w_1, w_2, \ldots, w_n\}$. Here, for each r $(1 \leq r \leq n)$, u_r, v_r, and w_r are the vertices corresponding to x_r. As for U, we initially set $U = \varnothing$. For all $1 \leq r, s \leq n$ such that $x_r x_s$ is an edge of G, we introduce into U a new vertex $\alpha_{r,s}$ (corresponding to the pair (r, s) and add the two edges $u_r \alpha_{r,s}$ and $\alpha_{r,s} v_s$ to G' (the 2-path $u_r, \alpha_{r,s}, v_s$ corresponds to the paths a, j, d and b, i, a in H). Note that when $x_r x_s$ is an edge of G, $x_s x_r$ is also an edge; hence, for pair (s, r) we add a new vertex $\alpha_{s,r}$.

For each pair v_r and w_r we add a new corresponding vertex β_r to U and add the edges $v_r \beta_r$ and $\beta_r w_r$ (corresponding to the paths d, j, a and a, k, e in H). Finally, for each pair u_r and w_r, we add a new vertex λ_r to U and then, add the two edges $u_r \lambda_r$ and $\lambda_r w_r$ to G'.

Defining the cost function: Define the cost function $c : V(G') \times V(H) \to \mathbb{Q}_{\geq 0}$ as follows. For each vertex $u_r \in V_1$ set $c(u_r, a) = 1$, $c(u_r, b) = 0$, and $c(u_r, l) = |G|$ for each $l \notin \{a, b\}$. For each vertex $v_r \in V_2$, set $c(v_r, a) = 1$, $c(v_r, d) = 0$, and $c(v_r, l) = |G|$ for each $l \notin \{a, d\}$. For each vertex $w_r \in V_3$, set $c(w_r, a) = 1$, $c(w_r, e) = 0$, and $c(w_r, l) = |G|$ for each $l \notin \{a, e\}$. Finally, for every $u \in U$, put $c(u, i) = c(u, j) = c(u, k) = 0$, and for every other case, set the cost to be $|G|$.

From a vertex cover in G to a homomorphism from G' to H: Let VC be a vertex cover in the original graph G. Define the mapping $f : V(G') \to V(H)$ as follows. For every vertex $u_r \in V_1$ set $f(u_r) = a$ if $x_r \in VC$; otherwise, set $f(u_r) = b$. For every $v_r \in V_2$ set $f(v_r) = a$ if $x_r \in VC$; otherwise, set $f(v_r) = d$. For every $w_r \in V_3$ set $f(w_r) = a$ if $x_r \notin VC$; otherwise, set $f(w_r) = e$. For every vertex $\alpha_{r,s}$ corresponding to pair (x_r, x_s) such that $x_r x_s \in E(G)$, set $f(\alpha_{r,s}) = i$ if $f(u_r) = b$; otherwise, set $f(\alpha_{r,s}) = j$. For every $\lambda_r \in G'$ where $u_r\lambda_r, \lambda_r w_r \in E(G')$, set $f(\lambda_r) = i$ if $f(u_r) = b$; otherwise, set $f(\lambda_r) = k$. Finally, for every $\beta_r \in G'$ with $v_r\beta_r, \beta_r w_r \in E(G')$, set $f(\beta_r) = j$ if $f(v_r) = d$; otherwise, set $f(\beta_r) = k$.

We show that f is a homomorphism from G' to H with cost $c(f) = |VC| + |G|$. Let $u_r\alpha_{r,s}$ be an edge of G'. Then, by the construction of G', $\alpha_{r,s}v_s$ is also an edge of G', where $\alpha_{r,s}$ corresponds to a pair (x_r, x_s) with $x_r x_s \in E(G)$. Since VC is a vertex cover for G, at least one of x_r and x_s is in VC. Without loss of generality, assume that $x_r \in VC$, and assume x_r corresponds to vertex u_r in V_1. Now, by definition, $f(u_r) = a$, and hence, $f(\alpha_{r,s}) = j$, where $aj \in E(H)$; thereby, $f(u_r)f(\alpha_{r,s}) \in E(H)$. Moreover, $f(v_s) \in \{a, d\}$, and hence, $f(\alpha_{r,s})f(v_s) \in E(H)$. Now, consider the edge $v_r\beta_r$ in G'. Notice that there is also an edge $\beta_r w_r$ of G' ($v_r \in V_2$, $w_r \in V_3$). First, consider the case where $x_r \notin VC$. Then, by definition, $f(w_r) = a$ and $f(v_r) = d$ and, consequently, $f(\beta_r) = j$; thus, $f(w_r)f(\beta_r) \in E(H)$, since aj is an edge of H. In this case, we additionally have $\beta_r v_r \in E(G')$, and, hence, $f(\beta_r)f(v_r) \in E(H)$. Now, consider the case where $x_r \in VC$. By definition, $f(v_r) = a$ and $f(w_r) = e$. In this case, we have $f(\beta_r) = k$ where β_r is the corresponding vertex in U to v_r and w_r. Since $ak, ek \in E(H)$, we have $f(v_r)f(\beta_r), f(\beta_r)f(w_r) \in E(H)$. A similar argument is applied when we consider a vertex $\lambda_r \in U$ corresponding to u_r and w_r. Therefore, f is a homomorphism from G' to H. It is easy to see that the cost of f is $|VC| + |VC| + |G| - |VC| = |G| + |VC|$.

From a homomorphism from G' to H to a vertex cover in G: Let f be a homomorphism from G' to H. To obtain a vertex cover in G, we modify f into a homomorphism so that it agrees on every $u_r \in V_1$ and $v_r \in V_2$. Suppose $f(u_r) = a$ and $f(v_r) = d$ for some $u_r \in V_1$ and $v_r \in V_2$. Consider the vertex $\beta_r \in U$ corresponding to v_r and w_r. Since v_r, β_r, w_r is a path in G', and there is no path of length two in H from d to e, we must have $f(w_r) = a$ and $f(\beta_r) = j$. Then, we define a homomorphism f' from G' to H as follows. We set $f'(v_r) = a$, $f'(w_r) = e$, and $f'(\beta_r) = k$. Moreover, for the vertex $\lambda_r \in U$ corresponding to vertices u_r and v_r, we set $f'(\lambda_r) = k$. Note that for every vertex $\alpha_{s,r}$ corresponding to a pair (x_s, x_r) with $x_r x_s \in E(G)$, we have $f(\alpha_{s,r}) = j$ and $f(u_s) = a$— notice that $\alpha_{s,r}v_r, u_s\alpha_{s,r} \in E(G')$. As such, we set $f'(\alpha_{s,r}) = i$, thereby, get $f(u_s)f'(\alpha_{s,r}) \in E(H)$. Finally, for every other vertex z, we set $f'(z) = f(z)$. It is easy to see that f' is a homomorphism from G' to H with $c(f) = c(f')$. Next, suppose for some u_s we have $f'(u_s) = b$ and $f'(v_s) = a$. By a similar modification, we modify f' further and obtain a homomorphism f'' so that $f''(u_s) = f''(v_s) = a$. We continue this process until we obtain a homomorphism f^t so that $f^t(u_r) = a$ if and only if $f^t(v_r) = a$ for every $1 \leq r \leq n$.

Therefore, for the sake of simplicity, we may assume $f^t = f$ and $f(u_r) = a$ if and only if $f(v_r) = a$ for every $u_r \in V_1$. Notice that if $f(u_r) = f(v_r) = a$, then we may assume $f(w_r) = e$. Otherwise, we change the image of w_r to e, and still, f is a homomorphism from G' to H, with a smaller cost.

Let $VC' = \{u_r, v_r \mid f(u_r) = f(v_r) = a\}$. Notice that as we discussed just above $VC' \cap \{u_s, v_s \mid f(w_s) = a\}| = \emptyset$. Therefore, $c(f) = |VC'| + |\{w_s \mid f(w_s) = a\}|$, and hence, $c(f) = |VC'| + |G| - \frac{|VC'|}{2}$. Let $VC = \{x_r \mid f(u_r) = a\}$, and notice that $|VC| = \frac{|VC'|}{2}$. Thus, $c(f) = |VC| + |G|$. We show that VC is a vertex cover in G. Suppose $x_r x_s \in E(G)$. Now there is a vertex $\alpha_{r,s} \in U$, and two edges $u_r\alpha_{r,s}, \alpha_{r,s}v_s$ in G'. Since, there is no path of length two between b, d in H and f is a homomorphism from G' to H, at least one of the $f(u_r), f(v_s)$ is a, say $f(u_r) = a$. Thus, by definition $u_r \in VC'$, and consequently $x_r \in VC$.

Showing the 1.128*-approximation is* ***NP****-hard:* We show that it is **NP**-hard to find a homomorphism $f : V(G') \to V(H)$ with $c(f) < (1 + \lambda)c(f^*)$ (here $\lambda = 0.128$, and f^* is the optimal minimum cost homomorphism from G' to H). For contradiction, suppose there is a polynomial-time algorithm that produces such a homomorphism f. Then, $c(f) = |VC|+|G|$ and $c(f^*) = |VC^*|+|G|$ (here VC^* is the optimal vertex cover in G). We have $|VC|+|G| < (1 + \lambda)(|VC^*| + |G|)$.

Thus, $|VC| < (1+\lambda)|VC^*|+\lambda|G|$, and hence, $|VC|-\lambda|G| < (1+\lambda)|VC^*|$. We may assume $|VC| \geq 0.639|G|$, thanks to the construction in [5]. Therefore, we have $|VC|(1 - \frac{\lambda}{0.639}) \leq |VC| - \lambda|G| < (1 + \lambda)|VC^*|$, and consequently, we have $|VC| < \frac{1+\lambda}{1-\frac{\lambda}{0.639}}|VC^*|$.

By setting $\frac{(1+\lambda)0.639}{0.639-\lambda} = \sqrt{2}$, we get a contradiction since, as shown in [25], the vertex cover cannot be approximated within any factor better than $\sqrt{2} - \epsilon$. Thus, $1 + \lambda = 1.128$ and it is NP-hard to approximate MinHOM(H) within factor 1.128 when H is a bipartite claw. Moreover, by setting $\frac{(1+\lambda)0.639}{0.639-\lambda} = 2$, ($\lambda = 0.242$) we get a contradiction with the $(2 - \epsilon)$-UG-hardness for the Vertex Cover [26]. That is, for every $\varepsilon \geq 0$, MinHOM(H) when H is a bipartite claw is 1.242-UG-hard.

Reduction for bipartite tent: Let $V_1 = \{u_1, u_2, \ldots, u_n\}$, $V_2 = \{v_1, v_2, \ldots, v_n\}$ and $V_3 = \{w_1, w_2, \ldots, w_n\}$ be three disjoint copies of $V(G) = \{x_1, x_2, \ldots, x_n\}$. Let set U be initially empty. At the end of the construction, the vertex set of G' is $V_1 \cup V_2 \cup V_3 \cup U$. For every edge $x_r x_s$ of G, we add the edges $u_r v_s$ and $v_s u_r$ into G'. For every $v_r \in V_2$ and $w_r \in V_3$, corresponding to vertex $x_r \in G$, add edge $v_r w_r$ into G'. Finally, for every $u_r \in V_1$ and $w_r \in V_3$, corresponding to vertex $x_r \in G$, add a new vertex λ_r to U, and add the edges $u_r\lambda_r$ and $\lambda_r w_r$ into G'. We define the cost function $c : V(G') \times V(H) \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ as follows. For every $u_r \in V_1$, set $c(u_r, a) = 1$, $c(u_r, b) = 0$, and $c(u_r, p) = |G|$ for every $p \notin \{a, b\}$. For every $v_r \in V_2$, set $c(v_r, j) = 1$, $c(v_r, l) = 0$, and $c(v_r, p) = |G|$ for every $p \notin \{l, j\}$. For every $w_r \in V_3$, set $c(w_r, a) = 1$, $c(w_r, d) = 0$, and $c(w_r, p) = |G|$ for every $p \notin \{a, d\}$. Finally, for every λ_r corresponding to vertices $u_r \in V_1$ and $w_r \in V_3$, set $c(\lambda_r, i) = c(\lambda_r, k) = 0$, and $c(\lambda_r, p) = |G|$ for every $p \notin \{i, k\}$. Now, by a similar argument as the one for the bipartite claw we get the inapproximability bound for MinHOM(H) when H is a bipartite tent.

Reduction for bipartite net: Let $V_1 = \{u_1, u_2, \ldots, u_n\}$, $V_2 = \{v_1, v_2, \ldots, v_n\}$ and $V_3 =$

$\{w_1, w_2, \ldots, w_n\}$ be three disjoint copies of $V(G) = \{x_1, x_2, \ldots, x_n\}$. Let sets U_1, U_2 be initially empty. At the end of the construction, the vertex set of G' is $V_1 \cup V_2 \cup V_3 \cup U_1 \cup U_2$. For every edge $x_r x_s$ of G, we add a new vertex $\alpha_{r,s}$ to U_1 and the edges $u_r\alpha_{r,s}, \alpha_{r,s}v_s$ into G' (here $u_r \in V_1$ is the copy of $x_r \in G$ and $v_s \in V_2$ is the copy of $x_s \in G$).

For every $v_r \in V_2$ and $w_r \in V_3$, corresponding to vertex $x_r \in G$, add edge $v_r w_r$ into G'. Finally, for every $u_r \in V_1$ and $w_r \in V_3$, corresponding to vertex $x_r \in G$, add two new vertices λ_r, β_r to U_2, and add the edges $u_r\lambda_r, \lambda_r\beta_r, \beta_r w_r$ into G'. We define the cost function $c : V(G') \times V(H) \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ as follows. For every $u_r \in V_1$, set $c(u_r, a) = 1$, $c(u_r, b) = 0$, and $c(u_r, p) = |G|$ for every $p \notin \{a, b\}$. For every $v_r \in V_2$, set $c(v_r, d) = 1$, $c(v_r, e) = 0$, and $c(v_r, p) = |G|$ for every $p \notin \{e, d\}$. For every $w_r \in V_3$, set $c(w_r, j) = 1$, $c(v_r, k) = 0$, and $c(v_r, p) = |G|$ for every $p \notin \{j, k\}$. For every $\alpha_{r,s} \in U_1$, set $c(\alpha_{r,s}, i) = c(\alpha_{r,s}, j) = 0$, and $c(\alpha_{r,s}, p) = |G|$ for every $p \notin \{i, j\}$. Finally, every $\lambda_r, \beta_r \in U_2$, corresponding to vertices $u_r \in V_1$ and $w_r \in V_3$, set $c(\lambda_r, a) = c(\lambda_r, d) = c(\beta_r, i) = c(\beta_r, j) = 0$ and for every other case the cost is $|G|$. Now, by a similar argument as the one for the bipartite claw, we get the inapproximability bound for MinHOM(H) when H is a bipartite net.

In conclusion, when H is a bipartite and MinHOM(H) is **NP**-complete, we get that MinHOM(H) is 1.128-approx-hard and 1.242-UG-hard.

***H* has vertices with self-loops:** We show that H must be reflexive; meaning every vertex has a loop. Otherwise, H must contain an induced sub-graph $H_1 = \{aa, ab\}$ where b does not have a self-loop (recall that we assume H is connected). As we mention in the introduction, Vertex Cover problem is an instance of MinHOM(H_1). Vertex Cover is $(\sqrt{2} - \epsilon)$-approx-hard and $(2 - \epsilon)$-UG-hard for every $\epsilon > 0$. Therefore, MinHOM(H_1) is $(\sqrt{2} - \epsilon)$-approx-hard and $(2 - \epsilon)$-UG-hard for every $\epsilon > 0$. By the hardness of approximation for sub-graphs (Lemma 5.1), we obtain better hardness bounds for MinHOM than the claim of the theorem. Therefore, we may assume that H is reflexive henceforth.

If H contains an induced cycle of length at least 4 (when removing the self-loops), LHOM(H) is **NP**-complete due to [6], and hence, by Lemma 5.4, MinHOM(H) does not admit any approximation. Thus, by Theorem 5.3 and Lemma 5.1, we need to consider the case where H is a claw, tent or net. When H is any of these three graphs, H contains an invertible pair (see Figure 5). In particular for the reflexive claw, we start with graph G as explained in the bipartite claw, and construct three partite graph G' with V_1, V_2, V_3, and we add an edge between $u_r \in V_1$ and $v_s \in V_2$ (corresponding to edges ae, aa, ba in the claw in Figure 5) if $x_r u_s \in E(G)$. Between $v_r \in V_1$ and $w_r \in V_2$ we place a path of length 2 (corresponding to walks a, d, d and a, d, a and e, e, a) and finally between $u_r \in V_1$ and $w_r \in V_3$ we add an edge. The cost function is defined as follows, $c(u_r, a) = 1$, $c(u_r, b) = 0$, for every $u_r \in V_1$, and $c(v_r, a) = 1$, $c(v_r, e) = 0$ for every $v_r \in V_2$. Finally for every $w_r \in V_3$, set $c(w_r, a) = 1$, $c(w_r, d) = 0$. The rest of the costs are defined in a similar way as in the bipartite claw reduction.

Now, by a similar argument for bipartite claw, we conclude that MinHOM(H) is 1.155-approx-hard and 1.389-UG-hard. Similar treatment is used for MinHOM(H) when H is a

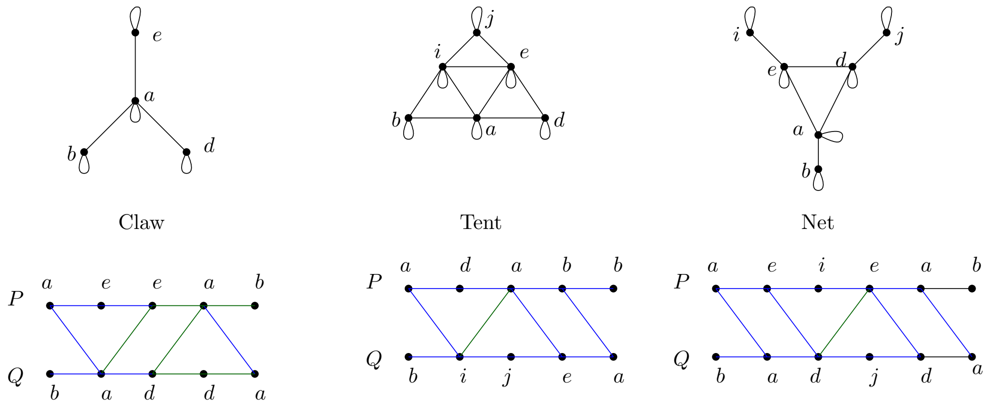


Figure 5: Invertible pair for claw, tent, and net.

reflexive net or a reflexive tent.

In conclusion, if H is reflexive and MinHOM(H) is **NP**-complete then MinHOM(H) is 1.155-approx-hard and 1.389-UG-hard. This completes the proof of the theorem.

□

References

[1] G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Channel assignment in wireless networks and classification of minimum graph homomorphism. *ECCC TR06*, 40, 2006.

[2] P. Austrin, S. Khot, and M. Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 74–80. IEEE Computer Society, 2009.

[3] A. Bar-Noy, M. Bellare, M. M. Halldórsson, H. Shachnai, and T. Tamir. On chromatic sums and distributed resource allocation. *Information and Computation*, 140(2):183–202, 1998.

[4] R. C. Brewster, T. Feder, P. Hell, J. Huang, and G. MacGillivray. Near-unanimity functions and varieties of reflexive graphs. *SIAM Journal on Discrete Mathematics*, 22(3):938–960, 2008.

[5] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.

[6] T. Feder and P. Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Series B*, 72(2):236–250, 1998.

[7] T. Feder, P. Hell, and J. Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999.

[8] T. Feder, P. Hell, and J. Huang. Bi-arc graphs and the complexity of list homomorphisms. *Journal of Graph Theory*, 42(1):61–80, 2003.

[9] T. Feder, P. Hell, J. Huang, and A. Rafiey. Adjusted interval digraphs. *Electronic Notes in Discrete Mathematics*, 32:83–91, 2009.

[10] A. Gupta, P. Hell, M. Karimi, and A. Rafiey. Minimum cost homomorphisms to reflexive digraphs. In E. S. Laber, C. F. Bornstein, L. T. Nogueira, and L. Faria, editors, *LATIN 2008: Theoretical Informatics, 8th Latin American Symposium, Búzios, Brazil, April 7-11, 2008, Proceedings*, volume 4957 of *Lecture Notes in Computer Science*, pages 182–193. Springer, 2008.

[11] G. Gutin, P. Hell, A. Rafiey, and A. Yeo. A dichotomy for minimum cost graph homomorphisms. *European Journal of Combinatorics*, 29(4):900–911, 2008.

[12] G. Z. Gutin, A. Rafiey, A. Yeo, and M. Tso. Level of repair analysis and minimum cost homomorphisms of graphs. *Discret. Appl. Math.*, 154(6):881–889, 2006.

[13] M. M. Halldórsson, G. Kortsarz, and H. Shachnai. Minimizing average completion of dedicated tasks and interval graphs. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, pages 114–126. Springer, 2001.

[14] P. Hell and J. Huang. Interval bigraphs and circular arc graphs. *Journal of Graph Theory*, 46(4):313–327, 2004.

[15] P. Hell, J. Huang, R. M. McConnell, and A. Rafiey. Interval-like graphs and digraphs. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS), 2018, August 27-31, 2018, Liverpool, UK*, pages 68:1–68:13, 2018.

[16] P. Hell, M. Mastrolilli, M. M. Nevisi, and A. Rafiey. Approximation of minimum cost homomorphisms. In *European Symposium on Algorithms (ESA)*, pages 587–598. Springer, 2012.

[17] P. Hell and J. Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.

[18] P. Hell and J. Nesetril. *Graphs and homomorphisms*. Oxford University Press, 2004.

[19] P. Hell and A. Rafiey. The dichotomy of list homomorphisms for digraphs. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1703–1713, 2011.

[20] P. Hell and A. Rafiey. The dichotomy of minimum cost homomorphism problems for digraphs. *SIAM Journal on Discrete Mathematics*, 26(4):1597–1608, 2012.

[21] P. Hell and A. Rafiey. Monotone proper interval digraphs and min-max orderings. *SIAM Journal on Discrete Mathematics*, 26(4):1576–1596, 2012.

[22] K. Jansen. Approximation results for the optimum cost chromatic partition problem. *J. Algorithms*, 34(1):54–89, 2000.

[23] T. Jiang and D. B. West. Coloring of trees with minimum sum of colors. *arXiv preprint math/9904140*, 1999.

[24] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 767–775. ACM, 2002.

[25] S. Khot, D. Minzer, and M. Safra. On independent sets, 2-to-2 games, and grassmann graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 576–589. ACM, 2017.

[26] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

[27] L. G. Kroon, A. Sen, H. Deng, and A. Roy. The optimal cost chromatic partition problem for trees and interval graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 279–292. Springer, 1996.

[28] M. Mastrolilli and A. Rafiey. On the approximation of minimum cost homomorphism to bipartite graphs. *Discrete Applied Mathematics*, 161(4):670–676, 2013.

[29] H. Müller. Recognizing interval digraphs and interval bigraphs in polynomial time. *Discret. Appl. Math.*, 78(1-3):189–205, 1997.

[30] A. Rafiey. Recognizing interval bigraphs by forbidden patterns. *J. Graph Theory*, 100(3):504–529, 2022.

[31] A. Rafiey, A. Rafiey, and T. Santos. Toward a dichotomy for approximation of h-coloring. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 91:1–91:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.