Bi-Arc Graphs and the Complexity of List Homomorphisms

Tomas Feder[*], Pavol Hell[†], and Jing Huang[‡]

June 3, 2002

Abstract

Given graphs G, H, and lists $L(v) \subseteq V(H), v \in V(G)$, a list homomorphism of G to H with respect to the lists L is a mapping $f : V(G) \rightarrow V(H)$ such that $uv \in E(G)$ implies $f(u)f(v) \in E(H)$, and $f(v) \in L(v)$ for all $v \in V(G)$. The list homomorphism problem for a fixed graph H asks whether or not an input graph G, together with lists $L(v) \subseteq V(H)$, $v \in V(G)$, admits a list homomorphism with respect to L.

In two earlier papers we classified the complexity of the list homomorphism problem in two important special cases: When H is a reflexive graph (every vertex has a loop), the problem is polynomial time solvable if H is an interval graph, and is NP-complete otherwise. When H is an irreflexive graph (no vertex has a loop), the problem is polynomial time solvable if H is bipartite and $\overline{H}$ is a circular arc graph, and is NP-complete otherwise.

In this paper we extend these classifications to arbitrary graphs H (each vertex may or may not have a loop). We introduce a new class of graphs, called bi-arc graphs, which contains both reflexive interval graphs (and no other reflexive graphs), and bipartite graphs with circular arc complements (and no other irreflexive graphs). We show that the problem is polynomial time solvable when H is a bi-arc graph, and is NP-complete otherwise. In the case when H is a tree (with loops allowed), we give a simpler algorithm based on a structural characterization.

1 Introduction

This paper deals with graphs in which loops are allowed. On the other hand, we do not consider parallel edges. (They are irrelevant from our point of view.) A graph in which

[*]tomas@Stanford.edu

[†]School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada, V5A 1S6; pavol@cs.sfu.ca

[‡]Department of Mathematics and Statistics, University of Victoria, P.O. Box 3045, Victoria, B.C., Canada, V8W 3P4; jing@math.uvic.ca

every vertex has a loop is called *reflexive*; a graph in which no vertex has a loop called *irreflexive*. In most situations, the definitions as applied to these graphs are the natural ones. For instance, a bipartite graph is, by definition, irreflexive. The only exception we shall make is the concept of a tree. We shall define, by convention, a tree in a way that allows loops: A *tree* is a connected graph without cycles of length greater than one. (Thus the standard trees shall be called irreflexive trees.)

A *homomorphism* f of a graph G to a graph H is a mapping $f : V(G) \to V(H)$ such that $uv \in E(G)$ implies $f(u)f(v) \in E(H)$. Note the role of vertices with loops: Firstly, if a vertex u has a loop in G, then $f(u)$ must also have a loop in H. Secondly, if u and v are adjacent in G and $f(u) = f(v)$, then the vertex $f(u)$ of G must have a loop. Homomorphisms to graphs in which loops are allowed are of interest in statistical physics, cf. [6, 7].

For a fixed graph H, the *homomorphism problem* HOMH asks whether or not an input graph G admits a homomorphism to H. When H is the complete irreflexive graph K_n, a homomorphism of G to H is simply an n-colouring of G, and hence HOMH is precisely the n-colouring problem. It is well-known that the n-colouring problem is NP-complete when $n > 2$ and polynomial time solvable when $n \leq 2$. The complexity of all homomorphism problems HOMH has been classified in [12]: HOMH is NP-complete if H is an irreflexive nonbipartite graph, and is polynomial time solvable otherwise.

Given graphs G, H, and lists $L(v) \subseteq V(H), v \in V(G)$, a *list homomorphism* of G to H with respect to the lists L is a homomorphism f of G to H, such that $f(v) \in L(v)$ for all $v \in V(G)$. For a fixed graph H, the *list homomorphism problem* LIST-HOMH asks whether or not an input graph G with lists L admits a list homomorphism to H with respect to L. As above, when $H = K_n$, a list homomorphism of G to H is simply a list colouring of G, and hence LIST-HOMH is the list colouring problem. The complexity of list colouring problems has been studied by J. Kratochvil and Z. Tuza [15]; among other results they have proved that the problem is polynomial time solvable when each list has at most two elements, but NP-complete even if each list is restricted to have at most three elements. List homomorphisms and list colourings also generalize problems of colouring graphs with some precoloured vertices, [1, 14].

A homomorphism of G to H is sometimes referred to as an *H-colouring* of G, and HOMH is referred to as the *H-colouring problem* [12]. In the same vein we could call a list homomorphism of G to H a *list H-colouring* (both with respect to some L) and also call LIST-HOMH the *list H-colouring problem.*

We have the following classifications of the complexity of LIST-HOMH in the reflexive and irreflexive cases:

Theorem 1.1 *[8] Let H be a reflexive graph. Then the problem* LIST-HOMH *is polynomial time solvable if H is an interval graph, and is NP-complete otherwise.* $\square$

Theorem 1.2 *[9] Let H be an irreflexive graph. Then the problem* LIST-HOMH *is* *polynomial time solvable if H is a bipartite graph whose complement $\overline{H}$ is a circular arc graph, and is NP-complete otherwise.* □

H is an *interval graph* if it admits an *interval representation*, i.e., a mapping that associates to each vertex x of H a real interval I_x so that two vertices x and y are adjacent in H if and only if I_x and I_y intersect. H is a *circular arc graph* if it admits a *circular arc representation*, whereby each vertex is associated with an arc on a circle, with two vertices adjacent if and only if the associated arcs intersect. Note that H is bipartite if and only if $\overline{H}$ has clique covering number two, thus the problem is polynomial time solvable for those graphs H whose complement is a circular arc graph of clique covering number two.

In this paper, we complete the classification of the complexity of all list homomorphism problems LIST-HOMH – without restrictions on H.

2 Bi-Arc Representations

The polynomial algorithms for LIST-HOMH in Theorems 1.1 and 1.2 take advantage of the geometric representation of H, i.e., the interval representation of a reflexive H, and the circular arc representation of the complement of an irreflexive H. With that in mind, we introduce a new kind of geometric representation.

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 both 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 can not contain bi-arcs $(N, S), (N', S')$ such that N intersects S' but S does not intersect N' or vice versa.

The class of bi-arc graphs contains all reflexive interval graphs and all complements of irreflexive circular arc graphs of clique covering number two. We first show how to obtain a bi-arc representation of a reflexive interval graph H from its interval representation (see Figure 1). Suppose first that H is reflexive and is represented by a family of intervals. Let C be a circle with two specified points p and q diametrically opposed on C. We transfer the family of intervals to a family of arcs by mapping each interval to an arc on the open segment extending from p to q clockwise. That is, each vertex v of H corresponds to an arc I_v from a point a_v to a point b_v clockwise in such a way that two *distinct* vertices v and v' are adjacent in H if and only if I_v intersects $I_{v'}$. We obtain a second set of arcs

J_v in a similar way by transfering the family of intervals to a family of arcs on the open segment extending from q to p. In the second set of arcs, each vertex v of H corresponds to an arc J_v, from a point c_v to a point d_v clockwise, in the open segment from q to p. Now our bi-arc representation of H will consist of pairs (N_v, S_v), with N_v being the arc from d_v to a_v, and S_v being the arc from b_v to c_v. Clearly, each N_v contains p but not q and each S_v contains q but not p. To see that this is a bi-arc representation of H, first note that for each vertex v (which is adjacent to itself) N_v is disjoint from S_v. Then consider two distinct vertices v and v' of H. If v and v' are adjacent, then I_v intersects $I_{v'}$ and J_v intersects $J_{v'}$. Since neither N_v nor $S_{v'}$ contains any point in the intersection of I_v and $I_{v'}$, N_v does not intersect $S_{v'}$; similarly, $N_{v'}$ does not intersect S_v. If v and v' are not adjacent, then I_v is disjoint from $I_{v'}$ and J_v is disjoint from $J_{v'}$. We see that N_v intersects $S_{v'}$ and $N_{v'}$ intersects S_v.

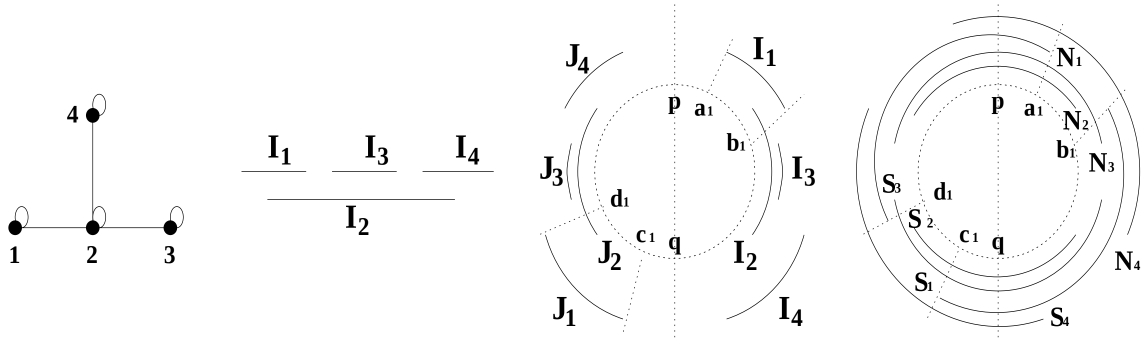

Figure 1:

Suppose now that H is a bipartite graph whose complement is a circular arc graph. Let (A, B) be a bipartition of H. Let C be a circle with the top point p, rightmost point r, bottom point q, and leftmost point s. According to [17] (see also [13]), H can be represented by a family $\mathcal{F}_1 = \{N_v : v \in A\} \cup \{S_v : v \in B\}$ of arcs on the circle satisfying the following properties: Each arc N_v contains both p and r, but neither q nor s, and each arc S_v contains both q and s, but neither p nor r. Further, two vertices $v \in A$ and $v' \in B$ are adjacent in H if and only if N_v and $S_{v'}$ are disjoint. Let $\mathcal{F}_2 = \{S_v : v \in A\} \cup \{N_v : v \in B\}$ be a new family of arcs, where each S_v with $v \in A$ (resp. N_v with $v \in B$) is obtained by reflecting N_v (resp. S_v) of $\mathcal{F}_1$ with respect to the line passing through r and s. Thus, in $\mathcal{F}_2$, each arc S_v contains both q and r, but neither p nor s, and each arc N_v contains p and s, but neither q or r. Further, two vertices $v \in A$ and $v' \in B$ are adjacent in H if and only if S_v and $N_{v'}$ do not intersect. These two families $\mathcal{F}_1$ and $\mathcal{F}_2$ together form a bi-arc representation of H (see Figure 2).

As we shall show in the next section, the reflexive bi-arc graphs are precisely the reflexive interval graphs and the irreflexive bi-arc graphs are precisely the complements of irreflexive circular arc graphs of clique covering number two. Hence, when H is reflexive or irreflexive, the list homomorphism problem LIST-HOMH is polynomial time solvable if H is a bi-arc graph and is NP-complete otherwise.

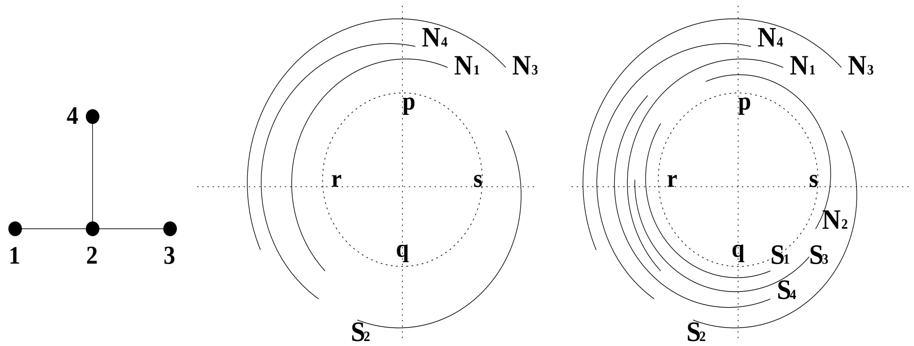

Figure 2:

The main result of this paper is that this holds for *all* graphs:

Theorem 2.1 *The list homomorphism problem* LIST-HOMH *is polynomial time solvable if* H *is a bi-arc graph, and is* NP*-complete otherwise.*

This will follow from Theorem 3.4 in the next section and Theorem 4.2 in Section 4.

3 NP-completeness

Given a graph H, the *associated bipartite graph* H^* is defined as follows. The vertex set of H^* consists of sets $A_H = \{x'; x \in V(H)\}$ and $B_H = \{x''; x \in V(H)\}$, forming a bipartition of H^*. The edge set of H^* consists of all edges $x'y''$ such that xy is an edge of H. Note that there are two kinds of edges in H^*, namely edges $x'x''$ where x has a loop in H, and edges $x'y''$ where $x \neq y$ and $xy \in E(H)$, which go in pairs, i.e., both $x'y''$ and $x''y'$ are present in H^*. (We observe that H^* is the 'categorical product' of H and K_2, cf. [4].)

A simple yet useful observation is that a bi-arc representation of H corresponds precisely to a circular arc representation of the complement of H^*. Thus,

Proposition 3.1 *H is a bi-arc graph if and only if the complement of H^* is a circular arc graph.* □

A *structure* $S = (V, E_1, E_2)$ consists of a finite set V, and two sets E_1 and E_2, each consisting of unordered pairs of elements from V. Each element of V is called a *vertex* and each unordered pair of E_1 (E_2) is referred to as a *red* (*blue*) edge of the structure S. Although each of (V, E_1) and (V, E_2) is a graph, the structure $S = (V, E_1, E_2)$ is *not* a graph. However, a structure can be viewed as an edge-coloured multigraph. If the vertex set V can be partitioned into A and B such that no red or blue edge has its both ends in A or in B, then the structure is called *bipartite* and (A, B) is called a *bipartition* of the structure. Given a graph H, the *associated structure* H^{**} has vertex set $V(H^{**}) = V(H^*)$, red edge set $E_1(H^{**}) = E(H^*)$, and blue edge set $E_2(H^{**}) = \{x'x'';\ x \in V(H)\}$. Clearly, H^{**} is bipartite with a bipartition (A_H, B_H).

Let $S = (V, E_1, E_2)$ be a fixed structure. The *structure list homomorphism problem* LIST-HOMS asks whether or not an input structure $Z = (V(Z), E_1(Z), E_2(Z))$, together with lists $L(v) \subseteq V, v \in V(Z)$, admits a mapping $f : V(Z) \to V$ such that $uv \in E_1(Z)$ implies $f(u)f(v) \in E_1$ and $uv \in E_2(Z)$ implies $f(u)f(v) \in E_2$. That is, each red edge is mapped to a red edge and each blue edge is mapped to a blue edge. We shall refer to such a mapping as a *structure list homomorphism* from $Z = (V(Z), E_1(Z), E_2(Z))$ to S, with respect to the lists L. Viewing a structure as an edge-coloured multigraph, a structure list homomorphism is simply a colour preserving list homomorphism. Colour preserving homomorphisms of edge-coloured multigraphs have been studied for instance in [2, 3].

Lemma 3.2 *For any graph H, the list homomorphism problem* LIST-HOMH *and the structure list homomorphism problem* LIST-HOMH^{**} *are polynomially equivalent.*

Proof: There is a natural way to transform an instance of LIST-HOMH to an instance of LIST-HOMH^{**} as follows: for each input G of LIST-HOMH with lists $L(v) \subseteq V(H)$, $v \in V(G)$, we let G^{**} be the associated structure of G defined above. To define the lists for G^{**}, let $L(v') = \{x'; x \in L(v)\}$ and $L(v'') = \{x''; x \in L(v)\}$. It is easy to see that G admits a list homomorphism to H if and only if G^{**} admits a structure list homomorphism to H^{**}.

We now show how to reduce LIST-HOMH^{**} to LIST-HOMH. Consider an instance (structure) $Z = (V(Z), E_1(Z), E_2(Z))$ of LIST-HOMH^{**}, together with lists $L(v) \subseteq V(H^{**})$, $v \in V(Z)$. Since H^{**} is bipartite, we may assume that Z is also bipartite with a bipartition (C, D). Further, we may assume that $L(v) \subseteq A_H$ for each $v \in C$, and $L(v) \subseteq B_H$ for each $v \in D$. (Recall that (A_H, B_H) is a bipartition of H^{**}. Thus each list in L consists of either primed or double-primed vertices.) We construct an instance of LIST-HOMH. Let G be the graph constructed from Z as follows: For each (nontrivial) component T induced by blue edges, contract T into a single vertex u and remove all blue loops resulting from the blue edges in T. The lists L_G of vertices of G are defined as follows: Firstly, we remove the primes and double primes from all the lists L in Z, i.e., replace each x' and x'' by x, in all lists in Z. Secondly, if u is contracted from T, we let $L_G(u) = \cap_{v \in V(T)} L(v)$ (the lists of uncontracted vertices do not change further). The construction of G and the lists L_G can be accomplished in polynomial time. It is easy to verify that Z admits a structure list homomorphism to H^{**}, with respect to the lists L, if and only if G admits a list homomorphism to H, with respect to the lists L_G. □

Lemma 3.3 *For any graph H, the list homomorphism problem* LIST-HOMH^* *can be reduced, in polynomial time, to the structure list homomorphism problem* LIST-HOMH^{**}.

Proof: Let $G = (V(G), E(G))$ be an input graph of LIST-HOMH^*, together with lists $L(v) \subseteq V(H^*)$, $v \in V(G)$. Let $Z = (V(Z), E_1(Z), E_2(Z))$ be the structure with

$V(Z) = V(G)$, $E_1(Z) = E(G)$, and $E_2(Z) = \emptyset$. It is obvious that G admits a list homomorphism to H^*, with respect to the lists L, if and only if Z admits a structure list homomorphism to H^{**}, with respect to the lists L. $\square$

Theorem 3.4 *The list homomorphism problem* LIST-HOMH *is NP-complete, unless H is a bi-arc graph.*

Proof: Suppose that H is not a bi-arc graph, i.e., that the complement of H^* is not a circular arc graph of clique covering number two. By Theorem 1.2, LIST-HOMH^* is NP-complete, and hence, by Lemmas 3.3 and 3.2, LIST-HOMH is also NP-complete. $\square$

An *edge-asteroid* in a bipartite graph, with the bipartition (X, Y), is a set of $2k+1$ edges $u_0v_0, u_1v_1, \ldots, u_{2k}v_{2k}$ ($k \geq 1$ and each $u_i \in X$ and $v_i \in Y$), and $2k+1$ paths, $P_{0,1}, P_{1,2}, \ldots, P_{2k,0}$, where each $P_{i,i+1}$ joins u_i to u_{i+1}, such that for each $i = 0, 1, \ldots, 2k$ there is no edge between $\{u_i, v_i\}$ and $\{v_{i+k}, v_{i+k+1}\} \cup V(P_{i+k,i+k+1})$. (Subscripts are modulo $2k+1$.) We refer to the (odd) integer $2k+1$ as the *order* of the edge-asteroid. We have the following characterization of circular arc graphs of clique covering number two:

Theorem 3.5 *[9] Let H be a bipartite graph. Then the complement of H is a circular arc graph if and only if H contains no induced cycles of length greater than four and no edge-asteroids.* $\square$

Corollary 3.6 *If H^* contains a cycle of length greater than four, or an edge-asteroid, then* LIST-HOMH *is NP-complete.*

Proof: This is implied by Theorems 3.4 and 3.5 together with the fact that H is a bi-arc graph if and only if the complement of H^* is a circular arc graph. $\square$

Theorem 3.7 *Let H be a reflexive graph. Then H is a bi-arc graph if and only if H is an interval graph.*

Proof: Sufficiency is shown in the previous section, so we only show necessity. Suppose that H is not an interval graph. By a well-known theorem of Lekkerkerker and Boland [16], H contains either an induced cycle of length greater than three or an asteroidal triple, i.e., a set of three vertices v_0, v_1, v_2, such that for each $i \in \{0, 1, 2\}$ there is a path joining v_i and v_{i+1} and avoiding the neighbours of v_{i+2} (subscripts are modulo 3). Assume first that $x_1x_2 \ldots, x_lx_1$ ($l > 3$) is an induced cycle of H. Then H^* contains an induced cycle of length greater than four (namely, $x'_1x''_1x'_2x''_3x'_4 \ldots x'_{l-2}x''_{l-1}x'_lx''_lx'_1$, when l is even, and $x'_1x''_1x'_2x''_3x'_4 \ldots x''l-2x'_{l-1}x''_lx'_1$, when l is odd). By Theorem 3.5, the complement of H^* is not a circular arc graph, and hence H is not a bi-arc graph. Assume that $\{v_0, v_1, v_2\}$ is an asteroidal triple in H. Let $x_{i_1}x_{i_2} \ldots x_{i_t}$ be a path joining v_i and v_{i+1} and avoiding the neighbours of v_{i+2}. Then $\{v'_0v''_0, v'_1v''_1, v'_2v''_2\}$ forms an edge-asteroid in H^*. Indeed, each $x'_{i_1}x''_{i_1}x'_{i_2}x''_{i_2} \ldots x'_{i_t}x''_{i_t}$ is a path joining $v'_iv''_i$ and $v'_{i+1}v''_{i+1}$ and avoiding the neighbours of $v'_{i+2}v''_{i+2}$. Again, by Theorem 3.5, the complement of H^* is not a circular arc graph, implying that H is not a bi-arc graph. $\square$

Theorem 3.8 *Let H be an irreflexive graph. Then H is a bi-arc graph if and only if it is the complement of a circular arc graph of clique covering number two.*

Proof: Again it only remains to show necessity. So assume that H is an irreflexive bi-arc graph. Suppose that H contains an induced odd cycle, say $x_1x_2\dots x_{2k+1}x_1$. Then H^* contains an induced cycle of length greater than four (namely, $x'_1x''_2x'_3\dots x''_{2k}x'_{2k+1}x''_1x'_2 \dots x''_{2k+1}x'_1$). By Theorem 3.5, the complement of H^* is not a circular arc graph, and hence H is not a bi-arc graph, contradicting the hypothesis. Hence H contains no induced odd cycles, and therefore it must be a bipartite graph. It follows that H^* is the union of two disjoint copies of H. Since H is a bi-arc graph, H^* is the complement of a circular arc graph of clique covering number two. Therefore the induced subgraph H of H^* is the complement of a circular arc graph of clique covering number two. $\square$

4 Polynomial Time Algorithms

Combining Theorems 3.7, 3.8, 1.1, and 1.2, we obtain the following fact:

Corollary 4.1 *When H is a reflexive or an irreflexive bi-arc graph,* LIST-HOMH *is polynomial time solvable.* $\square$

We proceed to prove that this remains so for all bi-arc graphs H.

Theorem 4.2 *When H is any bi-arc graph,* LIST-HOMH *is polynomial time solvable.*

We shall first consider the specialized algorithms for reflexive and irreflexive graphs. In [8, 9] we gave algorithms which use the geometric representation of H (by intervals or circular arcs) to encode LIST-HOMH as an instance of 2-satisfiability. We now describe an alternate approach.

Let H be an arbitrary graph. A function $g : V(H) \times V(H) \times V(H) \to V(H)$ is called a *majority choice function* if it satisfies the following properties:

1. $g(x, y, z) \in \{x, y, z\}$,
2. $g(x, w, w) = g(w, y, w) = g(w, w, z) = w$,
3. $g(x, y, z)g(x', y', z') \in E(H)$ whenever $xx', yy', zz' \in E(H)$.

Feder and Vardi [11] proved that if H admits a majority choice function, then LIST-HOMH is polynomial time solvable. In fact, they consider the following inference procedure [11]:

For each pair of distinct vertices $u, v \in V(G)$, let $S_{uv} = (L(u) \times L(v)) \cap E(H)$ if $uv \in E(G)$; otherwise let $S_{uv} = L(u) \times L(v)$. Here we assume that if $u \in V(G)$ has a loop, then every element of $L(u)$ also has a loop.

Next, repeatedly perform the following update operation for triples of distinct vertices $u, v, w \in V(G)$:

$S_{uv} \leftarrow \{ab \in S_{uv} : \exists c \in V(H), ac \in S_{uw} \text{ and } bc \in S_{vw}\}$.

If no set S_{uv} can be further reduced by an application of this operation, then the updating process terminates.

It is clear that if f is a list homomorphism of G to H with respect to the lists L, then at all times we must have $f(u)f(v) \in S_{uv}$ for all distinct $u, v \in V(G)$. In particular, if the sets S_{uv} become empty, then no list homomorphism exists.

On the other hand, in certain situations we may be able to conclude conversely, that if all sets S_{uv} are nonempty, a list homomorphism of G to H with respect to the lists L exists: We say that H has *strict width two* if, for all graphs G and lists L, whenever $f : U \to V(H)$, for some $U \subseteq V(G)$ with $|U| \geq 2$, is a mapping such that $f(u)f(v) \in S_{uv}$ for all distinct pairs of vertices $u, v \in U$, then there exists a list homomorphism $\hat{f}$ of G to H (with respect to the lists L) such that $\hat{f}(v) = f(v)$ for all $v \in U$. Feder and Vardi showed [11] that H admits a majority choice function if and only if it has strict width two. (The terminology of [11] applies the term 'strict width two' to the *problem* LIST-HOMH, rather than to the graph H as we do here. The results of [11] also apply in the more general context of constraint satisfaction problems.)

We now see that if H has strict width two, then the fact that all sets S_{uv} are nonempty, implies that a list homomorphism of G to H with respect to the lists L exists (and can be efficiently found): Choose some $U \subseteq V(G)$ with $|U| = 2$, say $U = \{u, v\}$, and choose some $f(u)f(v) \in S_{uv}$. Then repeatedly choose a single vertex $w \in V(G) - U$ and a vertex $c \in V(H)$ such that $f(u)c \in S_{uw}$ for all $u \in U$, so that we can proceed to add w to U by setting $f(w) = c$. The process terminates with $U = V(G)$ and a list homomorphism f.

We now explain how to use these techniques to give new polynomial time algorithms for finding list homomorphisms to a reflexive interval graph H, and to an irreflexive bipartite graph H whose complement is a circular arc graph.

Lemma 4.3 *If H is a reflexive interval graph, or an irreflexive bipartite graph whose complement is a circular arc graph, then H admits a majority choice function, i.e., has strict width two.*

Proof: Suppose first that H is a reflexive interval graph, and consider an interval representation of H. We shall define a majority choice function g on H. Consider $x, y, z \in V(H)$, and the intervals I_x, I_y, I_z representing them. Let I be the interval whose left endpoint is the middle point of the three left endpoints of I_x, I_y, I_z, and whose right endpoint is the middle point of the three right endpoints of I_x, I_y, I_z. One of the three intervals I_x, I_y, I_z, say I_t, must contain I, since two of the three intervals have left endpoint not to the right of the left endpoint of I, and two of the three intervals have right endpoint not to the left of the right endpoint of I. We define $g(x, y, z) = t$. (If two of x, y, z are the same vertex a, then we can make sure that t was chosen equal to a, since I_a will contain both the above middle points.)

It remains to show that if $xx', yy', zz' \in E(H)$, then $tt' \in E(H)$ for $t = g(x, y, z)$ and $t' = g(x', y', z')$. Suppose to the contrary that I_t is disjoint from $I_{t'}$, say I_t precedes $I_{t'}$. Then at least two of I_x, I_y, I_z precede at least two of $I_{x'}, I_{y'}, I_{z'}$, so either I_x is disjoint from $I_{x'}$, or I_y from $I_{y'}$, or I_z from $I_{z'}$, a contradiction.

If H is an irreflexive bipartite graph (with parts V_1, V_2), whose complement is a circular arc graph, the construction is similar. If two arguments are in V_1 and one in V_2, we can let g take the value of the first argument from V_1, and similarly if two arguments are in V_2 and one in V_1 let g take the value of the first argument from V_2. Thus it remains to define g when the three arguments are from the same part. Consider a circular arc representation of the complement of H. Vertices x in V_1 are represented by arcs N_x containing p but not q, vertices x' in V_2 by arcs $S_{x'}$ containing q but not p, and $x \in V_1$ is adjacent in H to $x' \in V_2$ if and only if N_x is disjoint from $S_{x'}$. When $x, y, z \in V_1$, we treat the arcs N_x, N_y, N_z as intervals in the line obtained by removing q, and choose t from x, y, z so that N_t is *contained in* the arc whose left endpoint is the middle point of the three left endpoints of N_x, N_y, N_z, and whose right endpoint is the middle point of the three right endpoints of N_x, N_y, N_z. We define $g(x, y, z) = t$, making sure once again that if two of x, y, z equal to a, then $t = a$. The definition of g when all three arguments x', y', z' are in V_2 is similar, by considering the corresponding intervals $S_{x'}, S_{y'}, S_{z'}$ containing q but not p. The proof that g is a majority choice function is analogous to the case of reflexive interval graphs. $\square$

Note that this lemma implies the existence of the polynomial algorithms in Theorems 1.1 and 1.2.

One may try to define a majority choice function for any bi-arc graph H; this however does not appear to be an easy task. We show instead, that H inherits strict width two from H^* (the associated bipartite graph of H).

Lemma 4.4 *If H^* has strict width two, then H also has strict width two.*

Proof: Let G be an instance of LIST-HOMH. Consider the final sets S_{uv} obtained with the above inference procedure, and suppose they are nonempty. Suppose we have a mapping $f : U \to V(H)$ for some $U \subseteq V(G)$ with $|U| \geq 2$ such that $f(u)f(v) \in S_{uv}$ for all pairs of distinct $u, v \in U$. To establish the strict width 2 property, we must show that there exists a list homomorphism $\hat{f}$ of G to H such that $\hat{f}(u) = f(u)$ for $u \in U$.

For distinct vertices $u, v \in V(G)$, let $S_u = \{a \in V(H) : \exists b \in V(H), ab \in S_{uv}\}$; note that S_u does not depend on the choice of v. Let G^* be the associated bipartite graph for G. If $u \in V(G)$, define lists for the corresponding vertices in G^* by $L^*(u') = \{a' : a \in S_u\}$, and $L^*(u'') = \{a'' : a \in S_u\}$. Treat G^* with these lists as an instance of LIST-HOMH^*, which has strict width 2, and obtain final sets S^*_{uv}. Note by comparison of the two inference procedures, that the sets S^*_{uv} are no more restrictive than the sets S_{uv}. That is, if $a \in S_u$, then $a'a'' \in S^*_{u'u''}$; if $ab \in S_{uv}$, then $a'b' \in S^*_{u'v'}$, $a'b'' \in S^*_{u'v''}$, $a''b' \in S^*_{u''v'}$, and $a''b'' \in S^*_{u''v''}$. This follows by induction on the successive updates of the S^*_{uv} when compared with the final S_{uv} (which satisfy their update rule with equality).

Consider the mapping f^* that sets $f^*(u') = a'$ and $f^*(u'') = a''$ whenever $f(u) = a$. By the above observation, this mapping satisfies the S^*_{uv} conditions, and since LIST-HOMH^* has strict width 2, there must exist a list homomorphism $\hat{f}^*_1$ of G^* to H^* that coincides with f^* where f^* is defined.

Say that a pair of vertices $u', u'' \in V(G^*)$ corresponding to $u \in V(G)$ is *matched* in $\hat{f}^*_1$ if whenever $\hat{f}^*_1(u') = a'$, we also have $\hat{f}^*_1(u'') = a''$. If all such pairs are matched in $\hat{f}^*_1$, then we can let $\hat{f}$ be the corresponding list homomorphism of G to H, that is, $\hat{f}(u) = a$ if $\hat{f}^*_1(u') = a'$ and $\hat{f}^*_1(u'') = a''$; this list homomorphism extends f as desired.

Suppose that not all pairs are matched in $\hat{f}^*_1$. We show how to select any one unmatched pair u', u'', and obtain a different list homomorphism $\hat{f}^*$ that has all the matched pairs from $\hat{f}^*_1$, with the same corresponding values as in $\hat{f}^*_1$, and in addition the pair u', u'' is also matched in $\hat{f}^*$. Repeating this process, we eventually get all pairs matched, and this gives as mentioned above the desired list homomorphism of G to H that extends f.

First, we define the list homomorphism $\hat{f}^*_2$ of G^* to H^* obtained from $\hat{f}^*_1$ by exchanging the two sides of the bipartite graphs G^*, H^*, by taking advantage of the symmetry of these two graphs and of the lists L^*. That is, if $\hat{f}^*_1(v') = b'$ and $\hat{f}^*_1(v'') = c''$, we let $\hat{f}^*_2(v') = c'$ and $\hat{f}^*_2(v'') = b''$. Note that $\hat{f}^*_1$ and $\hat{f}^*_2$ have the same matched pairs, with the same corresponding values.

Consider now the partial mapping that sets $f^*(u') = \hat{f}^*_1(u')$ and $f^*(u'') = \hat{f}^*_1(u'')$ for all matched pairs u', u'' in $\hat{f}^*_1$, and for some pair v', v'' as above that is not matched, sets $f^*(v') = \hat{f}^*_1(v') = b'$ and $f^*(v'') = \hat{f}^*_2(v'') = b''$. The mapping f^* satisfies all the S^*_{uv} conditions, since it satisfies those not involving v'' because of $\hat{f}^*_1$, those not involving v' because of $\hat{f}^*_2$, and the single extra condition $b'b'' \in S^*_{v'v''}$, as mentioned before, because $b \in S_v$.

Therefore, since LIST-HOMH^* has strict width 2, f^* can be extended to a list homomorphism $\hat{f}^*$ of G^* to H^* with at least one more matched pair than in $\hat{f}^*_1$, while the pairs that were already matched in $\hat{f}^*_1$ have the same values in $\hat{f}^*$. By induction, we eventually make all pairs be matched, and then $\hat{f}^*$ yields the desired list homomorphism $\hat{f}$ of G to H. This completes the proof of the lemma. □

If H is a bi-arc graph, then the associated bipartite graph H^* is the complement of a circular arc graph with clique covering number two. Thus H^* has strict width two, and therefore H has also strict width two, and hence LIST-HOMH is polynomial time solvable. This completes the proof of Theorem 4.2 and hence also of our main theorem. □

Corollary 4.5 *Every bi-arc graph admits a majority choice function.* □

We observe that if $P \neq NP$, then the converse also holds, as a graph H which is not a bi-arc graph yields an NP-complete problem LIST-HOMH, and hence H cannot have a majority choice function. In another paper [4] we prove this converse without assuming $P \neq NP$:

Theorem 4.6 *[4] A graph admits a majority choice function if and only if it is a bi-arc graph.* □

5 Bi-Arc Trees

Since each bi-arc graph H has strict width two, it must have a majority choice function. We do not know a direct construction of a majority choice function for an arbitrary bi-arc graph H. It is always possible to find such a function g for a given bi-arc graph H in polynomial time, if we so desire, by expressing the conditions on g as an instance of LIST-HOMH of size cubic in the size of H, and solving the problem by the above inference procedure. In this section we explicitly construct these majority choice functions for bi-arc *trees*. The construction relies on the structure of these trees, described in Theorem 5.3, and proved in a companion paper [10].

For a graph H, we use H^o to denote the subgraph of H induced by vertices which have loops. We say that H^o is *convex*, if every induced path P of H joining two vertices of H^o satisfies $V(P) \subseteq V(H^o)$.

Proposition 5.1 *If H is a bi-arc graph, then H^o is convex.*

Proof: Suppose that H^o is not convex. Then there is an induced path $x_1x_2 \ldots x_k$ ($k \geq 3$), such that $x_i \in V(H^o)$ if and only if $i = 1$ or k. Thus in H^* the vertices $x'_1, x''_1, x'_2, x''_2, \ldots, x'_k, x''_k$ form an induced cycle of length greater than four. By Theorem 3.5, the complement of H^* is not a circular arc graph and hence H is not a bi-arc graph. □

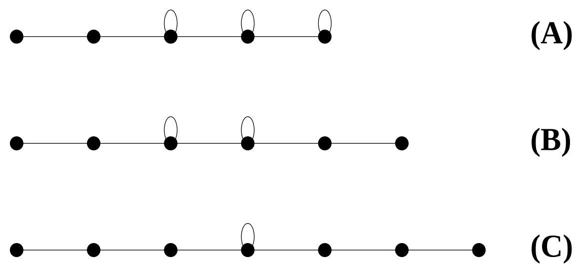

Figure 3:

In [10] we verify that if H is one of graphs in Figure 3 or 4, then H^* contains an edge-asteroid. Hence we have the following:

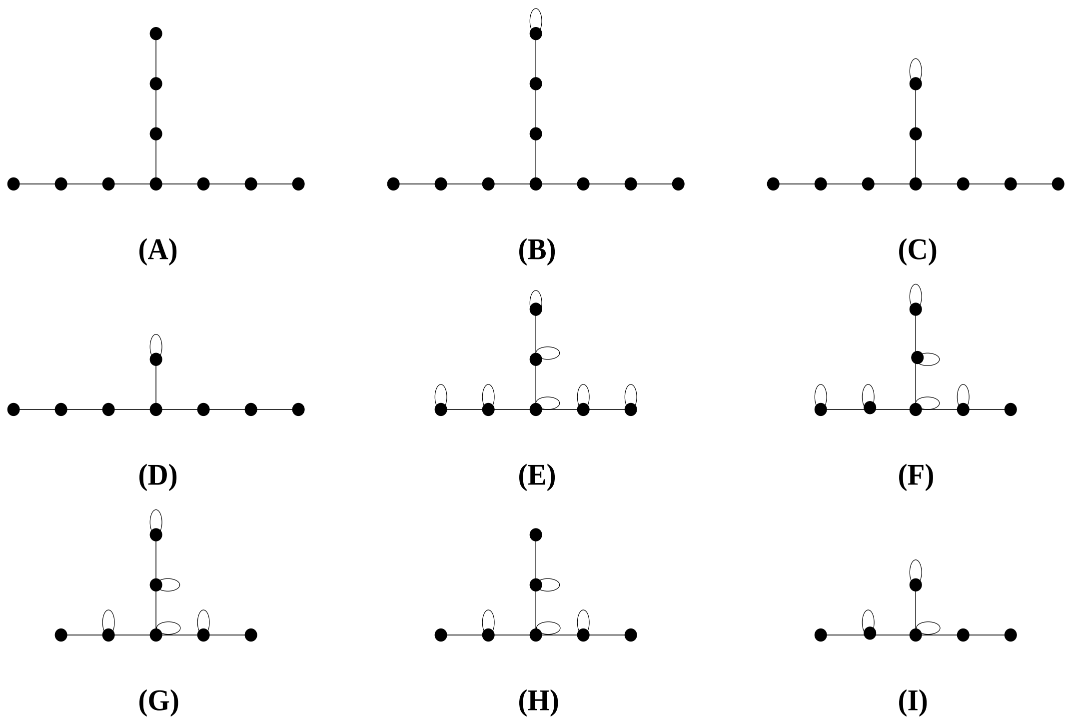

Figure 4:

Proposition 5.2 *[10] A bi-arc graph does not contain any graph from Figure 3 or from Figure 4 as an induced subgraph.* □

When H is a reflexive tree, H is an interval graph if and only if it does not contain the graph in Figure 4(E), [16]. Such a tree is also known as a *reflexive caterpillar*. Thus it follows from Theorem 3.7 and Corollary 4.1 that in the case when H is reflexive, LIST-HOMH is polynomial time solvable if H is a caterpillar, and is NP-complete otherwise. We shall refer to a reflexive caterpillar also as a *reflexive polynomial tree*. When H is an irreflexive tree, it follows from Theorem 3.5 that the complement of H is a circular arc graph if and only if H does not contain the graph in Figure 4(A) (since H does not contain any cycle and if it contains an edge-asteroid then it must contain the graph in Figure 4(A)). Hence, by Theorem 3.8 and Corollary 4.1, when H is irreflexive, LIST-HOMH is polynomial time solvable if H does not contain the graph in Figure 4(A), and is NP-complete otherwise. We shall refer to an irreflexive tree which does not contain the graph in Figure 4(A) as an *irreflexive polynomial tree*.

We define a vertex v in a tree to be a *good vertex* if there does not exist a path P of length six, with the middle vertex u, such that v is joined to u by a path (possibly of length zero) which is internally vertex-disjoint from P. For instance, every vertex of the graph in Figure 3(A) is a good vertex, and so is every vertex of the graph in Figure 3(B). All vertices except the center vertex of the graph in Figure 3(C) are good vertices. The graph in Figure 4(A) has no good vertices.

In [10] we prove that the above figures list all forbidden subtrees for polynomial trees. In fact, we have the following structure theorem:

Theorem 5.3 *[10] Let H be a tree such that H^o is a subtree of H. Then the following three statements are equivalent.*

(i) H is a bi-arc graph;

(ii) H does not contain as an induced subgraph any graph from Figure 3 or from Figure 4;

(iii) H is obtained either

from a reflexive polynomial tree by deleting the loops at a subset (possibly empty) of leaves, or

from an irreflexive polynomial tree by performing one of the following three operations:

- *adding no loops, or adding a loop on a good vertex v,*
- *adding a loop on a good vertex v and on a neighbour w of v, such that each neighbour of w other than v is a leaf,*
- *adding a loop on a good vertex v, and loops on any number of neighbours of v which are leaves.*

The cases are illustrated in Figures 5 and 6.

Recall that we have no explicit construction of the majority choice function for an arbitrary bi-arc graph (even though Corollary 4.5 guarantees that one exists). We shall now explicitly construct these majority choice functions for bi-arc trees, using the structure theorem, Theorem 5.3.

5.1 H is obtained from a reflexive polynomial tree

Suppose H is a bi-arc tree obtained from a reflexive polynomial tree by deleting the loops at a subset of leaves. We may assume that $V(H^o) \neq \emptyset$ because of Lemma 4.3. Let $P : v_1v_2 \ldots v_l$ be a longest path in H^o. Since H is a caterpillar, so is H^o. Each vertex of $H^o - V(P)$ and hence of $H - V(P)$ is a leaf of H. Let S_i be the set of vertices of $H - V(P)$ adjacent to $v_i \in V(P)$ for each $i = 1, 2, \ldots, l$. Thus for each $x \in V(H)$ there exists a unique $\pi(x)$ such that $x \in S_{\pi(x)} \cup \{v_{\pi(x)}\}$. For each i, $i = 1, 2, \ldots, l$, we let O_i be an ordering of the vertices of $S_i \cup \{v_i\}$, in which v_i is the first vertex.

We define a function $f : V(H) \times V(H) \times V(H) \to V(H)$ as follows: Let x, y, z be three vertices of H. If any two of x, y, z are the same vertex, say w, then $f(x, y, z) = w$; otherwise let m be the median of $\pi(x), \pi(y), \pi(z)$ and set $f(x, y, z)$ to be the first vertex of $\{x, y, z\}$ in the ordering O_m.

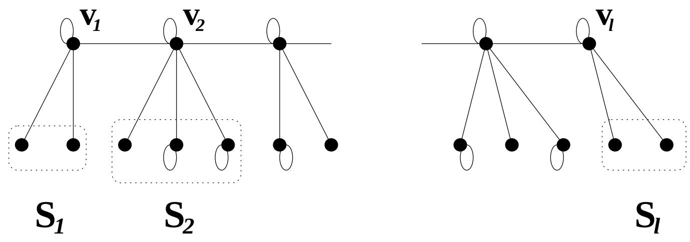

Figure 5:

Proposition 5.4 *The function f is a majority choice function.*

Proof: Clearly, f satisfies Properties 1 and 2 in the definition of a majority choice function. To verify Property 3, let xx', yy', zz' be three edges (each possibly a loop) of H. Let m be the median of $\pi(x), \pi(y), \pi(z)$, and m' the median of $\pi(x'), \pi(y'), \pi(z')$. Then we must have $|m - m'| \leq 1$: If, say $m \leq m' - 2$, then at least two of x, y, z have π values less than or equal to m and at least two of x', y', z' have π values greater than or equal to $m + 2$, which contradicts the fact that xx', yy', zz' are all edges of H.

Assume first that x, y, z are distinct, and x', y', z' are also distinct. If $v_m \in \{x, y, z\}$ and $v_{m'} \in \{x', y', z'\}$, then $f(x, y, z) = v_m$ is adjacent to $f(x', y', z') = v_{m'}$. If $v_m \notin \{x, y, z\}$ or $v_{m'} \notin \{x', y', z'\}$, then we must have $m = m'$. In this case, if $v_m \in \{x, y, z, x', y', z'\}$, then one of $f(x, y, z)$ and $f(x', y', z')$ is v_m and the other is in S_m. Hence $f(x, y, z)$ and $f(x', y', z')$ are adjacent. So assume that $v_m \notin \{x, y, z, x', y', z'\}$. It is then easy to see that $S_m \cap \{x, y, z\} = S_m \cap \{x', y', z'\}$ and each vertex of $S_m \cap \{x, y, z\}$ has a loop. This implies that $f(x, y, z)$ and $f(x', y', z')$ are adjacent.

Assume without loss of generality that two of x, y, z are the same vertex w. By definition, $f(x, y, z) = w$. If any two of x', y', z' are the same vertex w', then clearly $ww' \in E(H)$, i.e., $f(x, y, z)f(x', y', z') \in E(H)$. So assume that x', y', z' are all distinct. Let v be the first vertex in $O_{m'}$ from $\{x', y', z'\}$. By definition, $f(x', y', z') = v$. Since w is adjacent to at least two of x', y', z', w is adjacent to v, i.e., $f(x, y, z)f(x', y', z') \in E(H)$. $\square$

5.2 H is obtained from an irreflexive polynomial tree

Suppose that H is obtained from an irreflexive polynomial tree by performing one of the three operations listed in Theorem 5.3. We may again assume that H contains at least one loop. Let $Q : u_1 u_2 \ldots u_r$ be a longest path in H such that u_1 is the vertex v in $V^o(H)$ described in Theorem 5.3, and it is the only vertex in Q that has a loop. Note that the vertices of $H - V(Q)$ can again be partitioned into sets $T_1, T_2, \ldots, T_r$ such that $T_i \cup \{u_i\}$ induces a subtree for each $i = 1, 2, \ldots, r$. (Note that the maximality of Q assures that $T_r = \emptyset$.) Thus for each $x \in V(H)$ there is again a unique $\sigma(x)$ such that $x \in T_{\sigma(x)} \cup \{u_{\sigma(x)}\}$. Note that $\sigma(x) = 1$ for all $x \in V(H^o)$. Since v is a good vertex and Q is a longest path, the distance between u_i and any other vertex in $T_i \cup \{u_i\}$ is either one or two.

We order the vertices of H as follows: First for each $i = 1, 2, \ldots, r$ we obtain an ordering

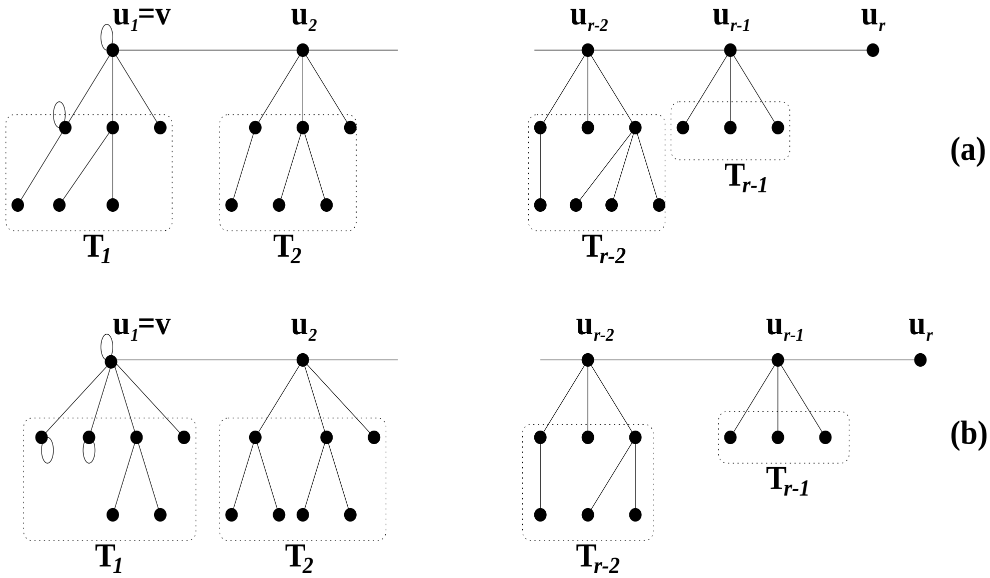

Figure 6:

O_i of the vertices of $T_i \cup \{u_i\}$, using the depth-first search starting at u_i. When $i = 1$, the search is in favour of vertices of H^o (note that $V(H^o) \subseteq T_1 \cup \{u_1\}$). Then we combine these orderings into a vertex ordering O of H by concatenating them in the order $O_1, O_2, \ldots, O_r$. Suppose that O is the ordering $v_1, v_2, \ldots, v_n$. Next, we colour the vertices of H with two colours so that two distinct adjacent vertices receive different colours. Call each of the two sets of vertices of the same colour a *colour class*.

We now define a function $g : V(H) \times V(H) \times V(H) \to V(H)$ as follows: Let x, y, z be three vertices of H.

(a) If any two of x, y, z are the same vertex, say w, then $g(x, y, z) = w$.

(b) Suppose that x, y, z are distinct and in the same colour class. Let m be the median of $\sigma(x)$, $\sigma(y)$, $\sigma(z)$. We define $g(x, y, z)$ to be the *first* vertex among x, y, z in the ordering O_m, except in the following four exceptional situations:

- All three vertices x, y, z lie in T_m with $m \geq 2$;
- All three vertices x, y, z lie in T_1 and at most one of them has a loop;
- Exactly two of x, y, z lie in T_1 and exactly one of them has a loop;
- Exactly two of x, y, z lie in T_1, neither has a loop, exactly one of them is adjacent to the unique neighbour of u_1 with a loop, and the third vertex of x, y, z is not u_1. (This last case only applies in the situation in Figure 6(a).)

In these situations, we define $g(x, y, z)$ be the *second* vertex among x, y, z in the ordering O_m.

(c) Suppose that x, y, z are distinct but not all in the same colour class. We define $g(x, y, z)$ to be the first vertex in O of the two vertices in the same colour class, except when $\{x, y, z\}$ contains u_1 and at least one of its leaf neighbours with a loop, in which case we define $g(x, y, z) = u_1$. (This exceptional case only applies in the situation in Figure 6(b).)

Proposition 5.5 *The function g is a majority choice function.*

Proof: To verify that g is a majority choice function, we need only show $g(x, y, z)g(x', y', z') \in E(H)$ when $xx', yy', zz' \in E(H)$.

Suppose that two of x, y, z are the same vertex w (and thus $g(x, y, z) = w$). If any two of x', y', z' are the same, then clearly $g(x, y, z)g(x', y', z') \in E(H)$. So assume x', y', z' are distinct. We first consider the case when x', y', z' are in the same colour class. Let m' be the median of $\sigma(x')$, $\sigma(y')$, $\sigma(z')$. Suppose that $m' \geq 2$ and $\{x', y', z'\} \subseteq T_{m'}$. By definition, $g(x', y', z')$ is the second vertex of $\{x', y', z'\}$ in $O_{m'}$. Since the vertex w is adjacent to at least two vertices of $\{x', y', z'\}$, it must be in $T_{m'} \cup \{u_{m'}\}$, and the depth-first search ordering $O_{m'}$ ensures that w is adjacent to $g(x', y', z')$. A similar discussion applies when $\{x', y', z'\} \subseteq T_1$ but at most one of x', y', z' has a loop. Suppose that exactly two of x', y', z' are in T_1. If exactly one of them has a loop, then both of them are neighbours of u_1 and we must have $w = u_1$; since w is adjacent to at least two vertices of x', y', z' it can not be the case that neither has a loop, exactly one is adjacent to the neighbour of u_1 with a loop, and $u_1 \notin \{x', y', z'\}$. Now in all other cases $g(x', y', z')$ is defined to be the first vertex of x', y', z' in $O_{m'}$. If $u_{m'} \in \{x', y', z'\}$, then $g(x', y', z') = u_{m'}$ and it is easy to see that w is adjacent to $u_{m'}$. So assume that $u_{m'} \notin \{x', y', z'\}$. If $m' \geq 2$ and $T_{m'}$ contains one or two vertices from $\{x', y', z'\}$, then $w \in T_{m'} \cup \{u_{m'}\}$ and w must be adjacent to the first vertex of $\{x', y', z'\}$ in the ordering $O_{m'}$, which is $g(x', y', z')$. If at least two of x', y', z' are in T_1 and both of them have loops, then they are both neighbours of u_1 and we must have that $w = u_1$. If exactly two of x', y', z' are in T_1, and neither has a loop or is adjacent to a neighbour of u_1 with a loop, then it is also easy to see that w is adjacent to the first vertex of x', y', z', which is $g(x', y', z')$. We now consider the case that x', y', z' are not all in the same colour class. Without loss of generality assume that y', z' are in the same colour class and that y' precedes z' in O. Then by definition $g(x', y', z') = x'$ or $g(x', y', z') = y'$. When $g(x', y', z') = x'$, it must be the case that $x' = u_1$ and y' is a leaf neighbour of u_1 with a loop. It is easy to see that w must be adjacent to $u_1 = g(x', y', z')$. On the other hand, when $g(x', y', z') = y'$, $wy' \in E(H)$, as otherwise at least one of x' and z' would have to have a loop, which is easy to see can not be the case.

Suppose that x, y, z are distinct and in the same colour class, and that x', y', z' are also distinct and in the same colour class. Observe that either $x = x'$, $y = y'$ and $z = z'$ or $\{x, y, z\}$ and $\{x', y', z'\}$ are in different colour classes. In the former case, each of x, y, z has a loop and clearly $g(x, y, z)g(x', y', z') \in E(H)$. For the latter case, assume first that $\sigma(x) < \sigma(y) < \sigma(z)$. It is easy to see that $\sigma(x') \leq \sigma(y') \leq \sigma(z')$. Since x', y', z' are distinct, either $\sigma(x') < \sigma(y')$ or $\sigma(y') < \sigma(z')$. Thus there are three possibilities: If $\sigma(x') < \sigma(y') < \sigma(z')$, then it is easy to see that $g(x, y, z)g(x', y', z') = yy' \in E(H)$. If $\sigma(x') < \sigma(y') = \sigma(z')$, then $\sigma(y) = \sigma(z) - 1$ and we have either $g(x,y,z)g(x',y',z') = yy' = u_{\sigma(y)}u_{\sigma(z)} \in E(H)$ or $g(x,y,z)g(x',y',z') = yz' = yu_{\sigma(y)} \in E(H)$. Finally, if $\sigma(x') = \sigma(y') < \sigma(z')$, then $\sigma(y) = \sigma(x) + 1$, and we have either $g(x,y,z)g(x',y',z') = yy' = u_{\sigma(y)}u_{\sigma(x)} \in E(H)$ or $g(x,y,z)g(x',y',z') = yx' = yu_{\sigma(y)} \in E(H)$. Assume then that $\sigma(x) < \sigma(y) = \sigma(z)$. Also assume that $g(x,y,z) = y$ (i.e., that y precedes z in $O_{\sigma(y)}$). If $y = u_{\sigma(y)}$, then y is adjacent to both $u_{\sigma(y)-1}$ and $u_{\sigma(y)+1}$, and also to all vertices of $T_{\sigma(y)}$ in the different colour class. It is easy to see that $g(x',y',z')$ must be one of these vertices and hence $g(x,y,z)g(x',y',z') \in E(H)$. If $y \neq u_{\sigma(y)}$, then $\sigma(y) = \sigma(y') = \sigma(z')$. If $u_{\sigma(y)} \in \{x',y',z'\}$, then $u_{\sigma(y)} = g(x',y',z')$ which must be adjacent to y; otherwise if $u_{\sigma(y)} \notin \{x',y',z'\}$, the depth-first ordering $O_{\sigma(y)}$ ensures that y is adjacent to the first vertex of $\{x',y',z'\}$ in $O_{\sigma(y)}$, which is $g(x',y',z')$. The case when $\sigma(x) = \sigma(y) < \sigma(z)$ can be treated similarly as for the case when $\sigma(x) < \sigma(y) = \sigma(z)$. So we assume that $\sigma(x) = \sigma(y) = \sigma(z)$. If $u_{\sigma(x)} \in \{x,y,z\}$, then $g(x,y,z) = u_{\sigma(x)}$ and $g(x',y',z')$ is one of vertices in $T_{\sigma(x)}$ adjacent to $u_{\sigma(x)}$. If $u_{\sigma(x)} \in \{x',y',z'\}$, then $g(x',y',z') = u_{\sigma(x)}$ is adjacent to all of x,y,z. If $u_{\sigma(x)} \notin \{x,y,z,x',y',z'\}$, then the middle vertex, say y, of $\{x,y,z\}$ is adjacent to the middle vertex y' of $\{x',y',z'\}$ in the ordering $O_{\sigma(x)}$.

Finally we consider the case when x,y,z are distinct but are not all in the same colour class. Suppose that $x = u_1$ and that z is a leaf neighbour of u_1 with a loop. Then $g(x,y,z) = x$. Then by analyzing all possible cases we see that $g(x,y,z)g(x',y',z') \in E(H)$. So we assume that x,y are in one colour class and z is in the other colour class. Assume further that $g(x,y,z) = x$ which precedes y in O. When x',y',z' are in the same colour class, either $x = x'$, $y = y'$, and $z \neq z'$ or $x \neq x'$, $y \neq y'$ and $z = z'$. In the former case, we have $g(x,y,z)g(x',y',z') = xx' \in E(H)$. If the latter case occurs, then either $z = z' = u_1$ and $g(x,y,z)g(x',y',z') = xz' \in E(H)$ or $z = z' \neq u_1$ and $g(x,y,z)g(x',y',z') = xx' \in E(H)$ (note that x' appears in the middle of $\{x',y',z'\}$ in O_1). Assume that x',y',z' are not all in the same colour class. There are three possibilities: Suppose that $x \neq x'$, $y \neq y'$ and $z \neq z'$. If x' appears before y', then $g(x,y,z)g(x',y',z') = xx' \in E(H)$; otherwise we must have that $\sigma(x) = \sigma(y)$ and $x = u_{\sigma(x)}$ and hence $g(x,y,z)g(x',y',z') = xy' \in E(H)$. Suppose that $x = x'$, $y \neq y'$ and $z \neq z'$. Then $g(x',y',z') = x'$ or $g(x',y',z') = u_1$ and we have $g(x,y,z)g(x',y',z') \in E(H)$. The case when $x \neq x'$, $y = y'$, and $z = z'$ does not occur because this means that y and z are adjacent vertices with loops and hence one of them is u_1, which contradicts the assumption that $g(x,y,z) = x$ (note also that x,y,z are distinct). Suppose that $x \neq x'$, $y = y'$, and $z \neq z'$. Then y has a loop and hence, by the assumption $g(x,y,z) = x$, x also has a loop. So both x and y are leaf neighbours of u_1. Therefore $g(x',y',z') = x' = u_1$ and $g(x,y,z)g(x',y',z') \in E(H)$. This completes the proof. $\square$

References

[1] M.O. Albertson and E.H. Moore, Extending graph colourings, J. Combinatorial Theory B 77 (1999) 83 - 95.

[2] N. Alon and T.H. Marshall, Homomorphisms of edge-coloured graphs and coxeter groups, J. Algebraic Combinatorics 8 (1998) 5 - 13.

[3] R.C. Brewster, The complexity of colouring symmetric relational systems, Discrete Applied Math. 49 (1994) 95 - 105.

[4] R.C. Brewster, T. Feder, P. Hell, J. Huang, and G. MacGillivray, Near-unanimity functions and varieties of graphs, in preparation.

[5] R.C. Brewster and P. Hell, Homomorphisms to edge-coloured cycles, in preparation.

[6] G.R. Brightwell and P. Winkler, Graph homomorphisms and phase transitions, J. Combinatorial Theory B 77 (1999) 221 - 262.

[7] G.R. Brightwell and P. Winkler, Gibbs measures and dismantlable graphs, J. Combinatorial Theory B 78 (2000) 141 - 166.

[8] T. Feder and P. Hell, List homomorphisms to reflexive graphs, J. Combinatorial Theory B 72 (1998) 236 - 250.

[9] T. Feder, P. Hell, and J. Huang, List homomorphisms and circular arc graphs, Combinatorica 19 (1999) 487 - 505.

[10] T. Feder, P. Hell, and J. Huang, Bi-arc graphs and the complexity of list homomorphisms, Preprint (2001) DMA-850-IR, Department of Math. and Stats., University of Victoria.

[11] T. Feder and M. Y. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: a study through datalog and group theory, SIAM J. Comput. 28 (1998) 57–104.]

[12] P. Hell and J. Nešetřil, On the complexity of H-colouring, J. Combinatorial Theory B 48 (1990) 92 - 110.

[13] P. Hell and J. Huang, Two remarks on on circular arc graphs, Graphs and Combinatorics 13 (1997) 65 - 72.

[14] J. Kratochvil and A. Sebo, Coloring precolored perfect graphs, J. Graph Theory 25 (1997) 207 - 215.

[15] J. Kratochvil and Z. Tuza, Algorithmic complexity of list colorings, Discrete Applied Math. 50 (1994) 297 - 302.

[16] C.G. Lekkerkerker and J.Ch. Boland, Representation of a finite graph by a set of intervals on the real line, Fund. Math. 51 (1962) 45 - 64.

[17] J. Spinrad, Circular-arc graphs with clique cover number two, J. Combinatorial Theory B (1988) 300 - 306.