SIAM J. DISCRETE MATH.
Vol. 22, No. 3, pp. 938–960

NEAR-UNANIMITY FUNCTIONS AND VARIETIES OF REFLEXIVE GRAPHS*

RICHARD C. BREWSTER†, TOMAS FEDER‡, PAVOL HELL§, JING HUANG¶, AND GARY MACGILLIVRAY¶

Abstract. Let H be a graph and $k \geq 3$. A *near-unanimity function of arity* k is a mapping g from the k-tuples over $V(H)$ to $V(H)$ such that $g(x_1, x_2, \ldots, x_k)$ is adjacent to $g(x'_1, x'_2, \ldots, x'_k)$ whenever $x_i x'_i \in E(H)$ for each $i = 1, 2, \ldots, k$, and $g(x_1, x_2, \ldots, x_k) = a$ whenever at least $k-1$ of the x_i's equal a. Feder and Vardi proved that, if a graph H admits a near-unanimity function, then the homomorphism extension (or retraction) problem for H is polynomial time solvable. We focus on near-unanimity functions on reflexive graphs. The best understood are reflexive chordal graphs H: they always admit a near-unanimity function. We bound the arity of these functions in several ways related to the size of the largest clique and the leafage of H, and we show that these bounds are tight. In particular, it will follow that the arity is bounded by $n - \sqrt{n} + 1$, where $n = |V(H)|$. We investigate substructures forbidden for reflexive graphs that admit a near-unanimity function. It will follow, for instance, that no reflexive cycle of length at least four admits a near-unanimity function of any arity. However, we exhibit nonchordal graphs which do admit near-unanimity functions. Finally, we characterize graphs which admit a conservative near-unanimity function. This characterization has been predicted by the results of Feder, Hell, and Huang. Specifically, those results imply that, if P $\neq$ NP, the graphs with conservative near-unanimity functions are precisely the so-called bi-arc graphs. We give a proof of this statement without assuming P $\neq$ NP.

Key words. graph homomorphism, near unanimity function, homomorphism extension, retraction, dichotomy

AMS subject classifications. 05Cxx, 68Q10, 68W25

DOI. 10.1137/S0895480103436748

1. Introduction.

We consider finite undirected graphs without multiple edges, but with loops allowed. A graph in which no vertex has a loop is called *irreflexive*, and a graph in which every vertex has a loop is called *reflexive*. When we say a graph satisfies a property, such as being connected, a tree, a cycle, etc., we mean that the underlying irreflexive graph (i.e., the graph obtained from it by deleting all loops if there are any) has the property.

Given graphs G and H, with lists $L(v) \subseteq V(H)$, for each $v \in V(G)$, a *list homomorphism* of G to H with respect to the lists L is a function $f : V(G) \to V(H)$ which satisfies the following two properties:

(i) $f(v) \in L(v)$ for all $v \in V(G)$;
(ii) $f(u)f(v) \in E(H)$ for all $uv \in E(G)$.

Note that a list homomorphism can map two adjacent vertices of G to the same vertex of H only if the vertex of H has a loop, and, in particular, it must map any vertex

*Received by the editors October 28, 2003; accepted for publication (in revised form) January 4, 2008; published electronically May 21, 2008. This work was partially supported by the Natural Science and Engineering Research Council of Canada.

http://www.siam.org/journals/sidma/22-3/43674.html

†Department of Math. and Stats., Thompson Rivers University, Kamloops, BC, Canada, V2C 3H5 (rbrewster@tru.ca).

‡Department of Computer Science, Stanford University, 268 Waverley St., Palo Alto, CA, 94301 (taras@theory.stanford.edu).

§School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, V5A 1S6 (pavol@cs.sfu.edu).

¶Department of Math. and Stats., University of Victoria, Victoria, BC, Canada, V8W 3P4 (jing@math.uvic.ca, gmacgill@uvic.ca).

of G with a loop to a vertex of H with a loop. List homomorphisms are introduced in [9].

For a fixed graph H, the *list homomorphism problem*, LIST-HOMH, asks whether an input graph G, together with lists L, admits a list homomorphism to H with respect to the given lists. The complexity of all list homomorphism problems has recently been classified [11]: LIST-HOMH is polynomial time solvable when H is a *bi-arc graph* and is NP-complete when H is not a bi-arc graph. (The definition of bi-arc graphs appears in section 5.)

In the case when the input lists $L(v) = V(H)$ for all $v \in V(G)$, the list homomorphism problem is the *homomorphism problem,* HOMH, or the *H-coloring problem.* When $H = K_n$, the irreflexive complete graph on n vertices, the homomorphism problem HOMH becomes the *n-coloring problem*, which is polynomial time solvable when $n \leq 2$ and NP-complete when $n \geq 3$. The complexity of all HOMH problems has been classified by Hell and Nešetřil [16]: HOMH is polynomial time solvable when H is bipartite or contains a loop and is NP-complete if H is irreflexive and not bipartite.

The *homomorphism extension problem*, EXTH, another special case of list homomorphisms, is of particular interest. In EXTH the inputs are restricted so that each list is either a singleton set or the entire set $V(H)$. Extension problems obviously correspond to questions of extending a given partial mapping ("precoloring") to a homomorphism and have been historically studied under an equivalent formulation called *retract problems,* RETH; cf. [7, 3, 15, 21, 30]. The *retract problem,* RETH, for a fixed graph H takes as an input a graph G containing H as a subgraph and asks whether or not there is a homomorphism f of G to H such that $f(v) = v$ for all $v \in V(H)$. Such a homomorphism f is called a *retraction* of G to H. If there is a retraction of G to H, then H is called a *retract* of G.

It seems difficult to classify the complexity of all extension problems. In particular, Feder and Vardi [13] have shown that extension problems capture the complexity of the much larger class of all constraint satisfaction problems (CSPs) in the following sense: for each CSP, say, Π, there exists a reflexive graph H such that Π and EXTH are polynomially equivalent. This means that even proving that each extension problem is NP-complete or is solvable in polynomial time would answer a difficult open question in complexity theory [13]. Recall that, by contrast, for list homomorphism problems, we know the exact classification of the complexity [9, 10, 11]. In particular, by techniques similar to [13] it can be seen that if a graph H admits a conservative near-unanimity function (as defined below), then LIST-HOMH is polynomial time solvable. In turn, this implies that RETH and EXTH are also polynomial time solvable. See [17].

Let H be a graph and $k \geq 3$ be an integer. A *near-unanimity function of arity* k (or NUF$_k$ for short) on H is a mapping $g : V(H)^k \to V(H)$ which satisfies the following properties:

(i) $g(x_1, x_2, \dots, x_k)$ is adjacent to $g(x'_1, x'_2, \dots, x'_k)$ whenever $x_i x'_i \in E(H)$ for each $i = 1, 2, \dots, k$, and

(ii) $g(x_1, x_2, \dots, x_k) = a$ whenever at least $k - 1$ of the x_i's equal a.

Early papers on near-unanimity functions include [1, 20]. Near-unanimity functions of arity 3, also called *majority functions*, are much studied [2, 17, 31]. A near-unanimity function g of arity k on H is a *conservative near-unanimity function* if $g(x_1, x_2, \dots, x_k) \in \{x_1, x_2, \dots, x_k\}$ for all vertices of H^k. It is shown in [11] that all bi-arc graphs admit a conservative near-unanimity function, implying that the corresponding list homomorphism problems can be solved in polynomial time via the results mentioned above. It is also shown in [11] that graphs H which are not bi-arc graphs have NP-complete list homomorphism problems. Hence, if we assume that $\mathrm{P} \neq \mathrm{NP}$, then bi-arc graphs are precisely the graphs which admit a conservative near-unanimity function. We will prove this is the case without the assumption $\mathrm{P} \neq \mathrm{NP}$.

The *categorical product* of a family of graphs, $\{H_i\}_{i \in I}$, denoted $\prod_{i \in I} H_i$, has as its vertex set the Cartesian product $\prod_{i \in I} V(H_i)$. (We restrict our attention to finite products, i.e., $|I| < \infty$.) Two vertices $(g_i)_{i \in I}$ and $(h_i)_{i \in I}$ are adjacent if g_i and h_i are adjacent in each $H_i, i \in I$. We may write $H_1 \times H_2 \times \cdots \times H_k$ for the product of the k graphs, $H_1, H_2, \ldots, H_k$, and H^k for the product of k copies of H. Thus, a near-unanimity function of arity k is a homomorphism of H^k to H that is *nearly unanimous*, i.e., satisfies condition (ii) above. Finally, a *graph variety* is a class $\mathcal{V}$ of graphs which contains all products and all retracts of members of $\mathcal{V}$. Given a class of graphs C, the *variety generated by* C is the smallest variety containing all of C. (The intersection of two varieties is itself a variety, and thus the concept of smallest is well defined.)

By abuse of notation we also denote by NUF_k the *class of all graphs that admit a* NUF_k. We let $\mathrm{NUF} = \bigcup_{k=1}^{\infty} \mathrm{NUF}_k$, i.e., the class of graphs each of which admits a near-unanimity function of some arity. We show that, for each fixed $k \geq 3$, the class NUF_k is a variety and that this collection of varieties is strictly monotone, i.e., $\mathrm{NUF}_3 \subset \mathrm{NUF}_4 \subset \cdots$ (with strict inclusions). We show the class of chordal graphs is contained in NUF. It follows that the *variety generated by chordal graphs*, i.e., the smallest variety containing all chordal graphs, is also contained in NUF; the variety generated by chordal graphs has been further investigated in [27]. The variety generated by *cop-win* or *dismantable* reflexive graphs (see [29]) contains the variety generated by chordal graphs. We give an example of a dismantable graph which does not belong to NUF. An extended examination of the inclusions described here (and of other varieties) is developed in [27]. In particular, NUF is strictly contained in the variety generated by dismantable graphs, and the variety generated by chordal graphs is strictly contained in the variety NUF [27, 25]. A polynomial time algorithm for recognizing graphs in NUF based on dismantability is given in [25].

We give two bounds on the arity of a near-unanimity function of a chordal graph with n vertices, in terms of its clique-size and its *leafage* (defined in section 3.1), respectively. It follows, in particular, that the arity is at most $n - \sqrt{n} + 1$. We present some forbidden substructures for graphs to have a NUF. It follows from these conditions that no reflexive cycle of length at least four admits a near-unanimity function. However, we shall exhibit nonchordal graphs which do admit near-unanimity functions. Finally, we give a proof, without assuming $\mathrm{P} \neq \mathrm{NP}$, of the result predicted by [11], that the graphs which admit a conservative near-unanimity function are precisely the bi-arc graphs. See also [5, 23].

2. Basic properties.

PROPOSITION 2.1. *For each $k \geq 3$, a graph H admits a NUF_k if and only if each connected component of H admits a NUF_k.*

Proof. Let $H_1, H_2, \ldots, H_p$ be the connected components of H. Suppose that $g : H^k \to H$ is a NUF_k on H. We begin by defining for each H_i a NUF_k, say, f_i, on H_i. Note that by definition g is a homomorphism of H^k to H, and hence any restriction $h = g|_X$ of g to a subgraph X of H^k is a homomorphism $h : X \to H$.

A vertex of H^k with at least $k-1$ coordinates equal is called a *nearly unanimous* vertex. For each connected component C of H_i^k, we define f_i on C as follows. If C has a nearly unanimous vertex $x = (x_1, x_2, \ldots, x_k)$, define $f_i = g|_C$. By near-unanimity, we have $f_i(x) \in H_i$. Since $f_i(C)$ is also connected, $f_i(C)$ is a subgraph of H_i. On the other hand, if C contains no such vertex, then define $f_i : C \to H_i$ by $f_i(z_1, z_2, \ldots, z_k) = z_1$. It is easily seen that each f_i is a NUF_k on H_i.

Conversely, suppose $g_i : H_i^k \to H_i$ is a NUF_k on H_i for each $i = 1, 2, \ldots, p$. Let C be a component of H^k, and let $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$ be two vertices in C. Since C is connected, there is a walk from x to y in C and, thus, a walk from x_s to y_s in H for each s, where $1 \leq s \leq k$. That is, for each component C of H^k and each coordinate s, there exists a component H_j of H such that $x_s \in H_j$ if and only if $y_s \in H_j$. Hence exactly one of the following conditions holds for all of the vertices of C:

(i) all k coordinates belong to the same H_j; i.e., C is a subgraph of H_j^k;

(ii) exactly $k-1$ coordinates belong to the same H_j, and the other coordinate belongs to H_m, $m \neq j$; or

(iii) at most $k-2$ coordinates belong to any H_j.

Let $x = (x_1, x_2, \ldots, x_k)$ be a vertex in C. In case (i), define $g(x) = g_j(x)$. In case (ii), choose some coordinate t such that $x_t \in H_j$. Define $g(x) = x_t$. (We fix t for the entire component C; thus, g is simply the projection of C onto its tth coordinate.) Finally, in case (iii), let $g(x) = x_1$.

In all cases, g is a homomorphism. Moreover, g satisfies the near-unanimity condition. In case (i), g inherits the property from g_j. In case (ii), any nearly unanimous vertex in C must have all $k-1$ coordinates from H_j, including x_t, equal. Finally, in case (iii) there are no nearly unanimous vertices. □

PROPOSITION 2.2. *For each $k \geq 3$, if a graph H admits a* NUF_k*, then H admits a* NUF_{k+1}.

Proof. Let $g : H^k \to H$ be a NUF_k on H. Then the function $h : H^{k+1} \to H$ defined as $h(x_1, x_2, \ldots, x_k, x_{k+1}) = g(x_1, x_2, \ldots, x_k)$ is a NUF_{k+1} on H. □

The argument above also proves the following.

COROLLARY 2.3. *For each $k \geq 3$, if a graph H admits a conservative* NUF_k*, then H admits a conservative* NUF_{k+1}.

We provide examples of graphs in $\mathrm{NUF}_{k+1} - \mathrm{NUF}_k$ for each $k \geq 3$ in section 4.1. Thus the converse of Proposition 2.2 does not hold. On the other hand, we will show that the converse of Corollary 2.3 does hold; i.e., H admits a conservative NUF_k for some $k \geq 3$ if and only if H admits a conservative NUF_3.

PROPOSITION 2.4. *Let G be a graph that admits a* NUF_k*, and let H be a retract of G. Then H also admits a* NUF_k.

Proof. Let g be a NUF_k on G, and let $r : G \to H$ be a retraction. Then it is easy to verify that $g' = r \circ (g\big|_{V(H)^k})$ is a NUF_k on H. □

The above result shows that the class of graphs which admit a NUF_k are closed under retractions. In the case of conservative functions, the class is also closed under taking induced subgraphs.

PROPOSITION 2.5. *Suppose the graph G admits a conservative* NUF_k *and H is an induced subgraph of G. Then H admits a conservative* NUF_k.

Proof. Let g be a conservative NUF_k for G. The restriction of g to H^k is a near-unanimity homomorphism of H^k to G, and to H as well since g is a conservative NUF. □

PROPOSITION 2.6. *Let $X_1, X_2, \ldots, X_n$ be graphs in* NUF_k*. Then the product $X_1 \times X_2 \times \cdots \times X_n$ is also in* NUF_k.

Proof. By associativity of the product, it suffices to verify the claim for two graphs. It is easy to check that if g_X and g_Y are near-unanimity functions of arity k for X

and Y, respectively, then

$$g((x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)) = (g_X(x_1, x_2, \ldots, x_k), g_Y(y_1, y_2, \ldots, y_k))$$

is a NUF_k for $X \times Y$. ◻

COROLLARY 2.7. *For each $k \geq 3$, the class NUF_k is a variety.*

A *dominating vertex* of a graph G is one adjacent to all other vertices of G. We make the following observation.

PROPOSITION 2.8. *If the reflexive graph H has a dominating vertex v, then H admits a NUF_k for all $k \geq 3$.*

Proof. By Proposition 2.2 it suffices to show that H admits a NUF_3. The function

$$g(a, b, c) = \begin{cases} x & \text{if at least two of } a, b, c \text{ equal } x, \\ v & \text{otherwise} \end{cases}$$

is a NUF_3. ◻

A graph is *chordal* if it does not contain an induced cycle of length greater than three. We shall show that each chordal graph belongs to NUF, and we shall show that each reflexive cycle of length at least four does not. However, Proposition 2.8 allows us to find examples of nonchordal graphs with a NUF. The *wheel* w_n is the graph obtained from a cycle of length n by adjoining one (new) vertex adjacent to all other vertices.

COROLLARY 2.9. *Each reflexive wheel $w_n, n \geq 4$, is a nonchordal graph in NUF_3.*

Since NUF_3 is the variety generated by finite paths [2, 17, 19], we see that each reflexive wheel is in the variety generated by finite paths and thus in the variety generated by chordal graphs. In [27] it is shown that the variety generated by chordal graphs is in fact strictly contained in the variety NUF.

3. Reflexive chordal graphs.

In this section and the next section, all graphs are assumed to be reflexive unless otherwise stated. We show in this section that every reflexive chordal graph admits a near-unanimity function. (By contrast, in section 4.2 we shall show that no reflexive cycle of length greater than three admits a near-unanimity function.) We provide two bounds on the arity of the NUF. One bound is based on the *leafage* of the graph, and the second is based on the clique-size and is obtained through the study of *tree obstructions*. We remark that the two approaches, leafage and tree obstructions, often allow us to compute the minimum k for which a graph admits a NUF_k. Typically the leafage is used to demonstrate the existence of a NUF_k, and tree obstructions are used to prove the nonexistence of a NUF_{k-1}.

3.1. Arity bounds based on leafage.

Let T be a tree. A *subtree* of T is a connected subgraph of T. A *rooted subtree* R of T is a subtree of T with a distinguished vertex called the *root* of R, denoted $r(R)$.

It is well known that a graph H is chordal if and only if it is the *intersection graph* of a family $\mathcal{F}$ of subtrees of a tree; that is, there is a one-to-one correspondence between $V(H)$ and $\mathcal{F}$ such that two vertices of $V(H)$ are adjacent in H if and only if the corresponding subtrees of $\mathcal{F}$ have at least one vertex in common; see [14]. The family $\mathcal{F}$, together with the underlying tree, is called an *intersection representation* of H by subtrees. The *leafage* $l(H)$ of a chordal graph H is the minimum number of leaves of a tree in which H has an intersection representation; cf. [26].

In the following H is a chordal graph. We use $\mathcal{F}$ to denote a (fixed) family of subtrees of a tree T which gives an intersection representation of H. Further, let $\mathcal{T}$

be the set of all rooted subtrees R of T such that $R-r(R)$ is the union of some (zero, one, or more) components of $T-r(R)$, and if $R-r(R)$ is the union of zero components of $T-r(R)$, then $r(R)$ is a leaf of T. For a rooted subtree R, we denote by l_R the number of leaves of T contained in R; note that these definitions ensure that $l_R \geq 1$ for any rooted subtree R.

Given a collection S of $k+1$ (not necessarily distinct) subtrees in $\mathcal{F}$, we say that a rooted subtree $R \in \mathcal{T}$ is *critical with respect to* S if it satisfies the following two properties:

1. there are at least $k - l_R + 1$ subtrees of S, each of which contains a vertex of R;
2. for each $R' \in \mathcal{T}$ contained in $R - r(R)$, there are at most $k - l_{R'}$ subtrees of S, each of which contains a vertex in R'.

The two conditions above are referred to as Properties 1 and 2.

LEMMA 3.1. *Let S be a family of $k+1$ (not necessarily distinct) members of $\mathcal{F}$. Then*

(a) *every rooted subtree in $\mathcal{T}$ satisfying Property* 1 *contains a rooted subtree which is critical with respect to S; and*

(b) *there are pairwise vertex disjoint critical rooted subtrees $R_1, \ldots, R_p$ such that $T - \bigcup_{i=1}^{p} R_i$ does not contain any critical rooted subtrees.*

Proof. Let the rooted subtree X satisfy Property 1. If X also satisfies Property 2, then X is critical with respect to S, and we are done. Otherwise, $X - r(X)$ contains another rooted subtree X' which satisfies Property 1. Again, if X' also satisfies Property 2, then X' is critical; otherwise, $X' - r(X')$ contains a third rooted subtree X'' which satisfies Property 1. Continuing this way, we will find a critical subtree with respect to S. A subtree consisting of a single vertex which satisfies Property 1 trivially satisfies Property 2.

To see statement (b), note that the entire tree T (with an arbitrary root) satisfies Property 1. In addition, a rooted tree R which does not satisfy Property 1 cannot contain a subtree R' which satisfies Property 1, since $k - l_R + 1 \leq k - l_{R'} + 1$ in the case that R' is a subtree of R. ∎

THEOREM 3.2. *Every chordal graph H of leafage k admits a* NUF_{k+1}.

Proof. Let $\mathcal{F}$ be an intersection representation of H by subtrees of a tree T with $k \geq 2$ leaves. We shall show that the intersection graph H of $\mathcal{F}$ admits a NUF_{k+1}. By Proposition 2.1, we may assume that H is connected. Further, we assume that every vertex of T belongs to a subtree in $\mathcal{F}$. For convenience, we shall not distinguish between the vertices of H and the subtrees of $\mathcal{F}$.

Let S be a collection of $k+1$ subtrees of $\mathcal{F}$. Although some of the $k+1$ subtrees of S may be the same, we treat them as distinct in the counting below.

By Lemma 3.1, there exist $R_1, R_2, \ldots, R_p$ pairwise vertex-disjoint critical rooted subtrees with respect to S such that $T - \cup_{i=1}^{p} R_i$ does not contain any critical rooted subtrees. We claim that there is a subtree in $\mathcal{F}$ containing all the roots $r(R_i)$, $i = 1, 2, \ldots, p$. When $p = 1$, this is clearly true as by assumption every vertex of T is in a subtree of $\mathcal{F}$. So assume that $p \geq 2$. In this case, we prove the stronger statement that in fact there is a subtree in S which contains all the roots. Suppose to the contrary that none of the $k+1$ subtrees of S contains all the roots. Then each subtree of S has vertices in at most $p-1$ critical rooted subtrees R_i. (Observe that any subtree of S that contains vertices in any two rooted subtrees must in fact contain both roots of the subtrees.) Denote by c_i ($i = 1, 2, \ldots, p$) the number of subtrees of S, each of which has a vertex in R_i. Then we have $(p-1)|S| = (p-1)(k+1) \geq \sum_{i=1}^{p} c_i$.

By Property 1, $c_i \geq k - l_{R_i} + 1$. Thus we have

$$\begin{aligned}(p-1)(k+1) &\geq \sum_{i=1}^{p} c_i \\ &\geq \sum_{i=1}^{p}(k - l_{R_i} + 1) \\ &= p(k+1) - \sum_{i=1}^{p} l_{R_i},\end{aligned}$$

which gives $\sum_{i=1}^{p} l_{R_i} \geq (k+1)$. This implies that some leaf of T must be contained in at least two critical rooted subtrees, contradicting the assumption that the critical rooted subtrees are pairwise vertex-disjoint.

Ultimately the goal is to define a NUF on these families of subtrees. Suppose at least k subtrees of S are the same subtree X. We claim X contains all the roots. In fact, a rooted subtree $R \in \mathcal{T}$ is critical with respect to S if and only if $V(R) \cap V(X) = \{r(R)\}$. To see this, observe that R satisfying Property 1 requires $V(X) \cap V(R) \neq \emptyset$ and R satisfying Property 2 ensures $V(X) \cap V(R - r(R)) = \emptyset$. On the other hand, $V(X) \cap V(R) = \{r(R)\}$ implies that R contains vertices from at least $k \geq k - l_R + 1$ elements of S, since $l_R \geq 1$. Also, $R - r(R)$ contains no vertices from the k copies of X, which implies that $R - r(R)$ contains vertices from at most $k - l_{R-r(R)}$ subtrees of S, as $l_{R-r(R)} \geq 1$. Consequently, X contains all the roots.

It remains to define the near-unanimity function g on H. Given $k+1$ vertices of H, consider the corresponding family S of subtrees in $\mathcal{F}$. Decompose T into p critical rooted subtrees $\{R_i\}_{i=1}^{p}$ as described in Lemma 3.1. Define $g(S)$ as follows: When at least k subtrees of S are the same subtree X, let $g(S) = X$; otherwise, let $g(S)$ be any subtree of $\mathcal{F}$ which contains all the roots $r(R_i)$. It remains to verify that g is a homomorphism from H^{k+1} to H. Thus consider two adjacent vertices in H^{k+1}. That is, let $S = \{U_1, U_2, \ldots, U_{k+1}\}$ and $S' = \{V_1, V_2, \ldots, V_{k+1}\}$ be two collections of $k+1$ subtrees (from $\mathcal{F}$) such that U_j intersects with V_j for each $j = 1, 2, \ldots, k+1$. Again, let $R_1, R_2, \ldots, R_p$ be the decomposition of T into critical subtrees *with respect to* S. Suppose to the contrary that $g(S)$ and $g(S')$ do not intersect. Let $P : z_0 z_1 \ldots z_d$ be the shortest path from $g(S)$ to $g(S')$ where $z_0 \in V(g(S))$ and $z_d \in V(g(S'))$. Let C be the component of $T - z_d$ containing z_0 and C' be the component of $T - z_0$ containing z_d. Let A be the rooted subtree consisting of C with $r(A) = z_{d-1}$. Then A is a rooted subtree containing $g(S)$ but no vertex from $g(S')$. Similarly, $B = C'$ with $r(B) = z_1$ is a rooted subtree containing $g(S')$ but no vertex from $g(S)$. Since each leaf of T is either in A or in B, we must have $l_A + l_B \geq k$. Since $g(S)$ does not intersect with B, B contains none of the roots of $R_1, R_2, \ldots, R_p$. Thus, B cannot satisfy Property 1 with respect to S. This means that S contains at most $k - l_B$ subtrees such that each of them has a vertex in B. In other words, S contains at least $k+1-(k-l_B) = l_B + 1$ subtrees, none of which has a vertex in B. Thus, these $l_B + 1$ subtrees must all be contained in A. Each of these subtrees intersects a member of S'. Hence S' must contain at least $l_B + 1$ subtrees, each of which contains a vertex in A. On the other hand, a similar argument shows that S' contains at most $k - l_A$ subtrees, each of which has a vertex in A. So we must have $l_B + 1 \leq k - l_A$, i.e., $l_A + l_B \leq k - 1$, in contradiction to the fact that $l_A + l_B \geq k$. ∎

Lin, McKee, and West [26] proved that for every chordal graph H with n vertices, the leafage is at most $n - \lg n - \frac{1}{2} \lg \lg n + O(1)$. Hence, each chordal graph with n

vertices admits a NUF of arity $n - \lg n - \frac{1}{2}\lg\lg n + O(1)$. We improve this bound in the next section.

The upper bound on the arity in Theorem 3.2 is sharp in the sense that there are chordal graphs of leafage l which do not admit a NUF_l. In section 4.1, we construct families of graphs useful for showing lower bounds, including the one just mentioned.

3.2. Arity bounds based of tree certificates. We recall the problem EXTH from the introduction. An instance of EXTH consists of a graph G together with lists $L(v) \subseteq V(H)$, where each $L(v)$ is either a singleton set or all of $V(H)$. Let $X \subseteq V(G)$ be the set of vertices x for which $L(x)$ is a singleton. Then we may view the function $p : X \to V(H)$ defined by $p(x) = h \in L(x)$ as a preassignment of images (in H) to the vertices in X. The vertices in X are called *preassigned* vertices. The EXTH problem asks if there exists a homomorphism $f : G \to H$ which extends the preassignment p (i.e., satisfies $f(x) = p(x)$ for $x \in X$). In the case of a yes instance, we say that p is *extendible (in H).* In the following we shall use the language of lists, or of extending preassignments, as is convenient.

Before entering the technical details of our work, we outline some key ideas used in the development. (We use the standard notation for $X \subseteq V(G)$, and $G[X]$ denotes the induced subgraph of G with vertex set X.) First, it is clear that, given a preassignment $p : X \to V(H)$, if p is not a homomorphism of $G[X]$ to H, then p is not extendible. On the other hand, if p is a homomorphism $G[X] \to H$, then a natural algorithmic idea is to successively extended p by one vertex, i.e., select $v \in V(G)\backslash X$, and define $f(v) = h \in L(v)$ (and set $f(u) = p(u)$ for all $u \in X$) such that $f : G[X \cup \{v\}] \to H$ is a homomorphism. Clearly the condition we must verify is that $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$ for each $u \in X$. Such a value h is an *allowed image* for v. On the other hand, in searching $L(v)$ for an allowed image for v, we may remove from $L(v)$ any value h' such that $p(u)h' \notin E(H)$ for some $u \in X$, where $uv \in E(G)$. Below we will talk about u *causing* h' to be removed from $L(v)$. Finally, if some $L(v)$ becomes empty by removing nonallowed images, then $p : X \to V(H)$ cannot extend to a homomorphism of G to H. (The image of v under any homomorphism $\phi : G \to H$ extending p will always be an allowed image, and thus $\phi(v)$ is never removed from $L(v)$.) This process of removing nonallowed images is known as a *consistency check*; see, for example, [17].

Testing all possible extensions of p to all of G is an exponential process; however, for certain graphs H, as identified below, the extendability of p can be determined by considering only a polynomial number of possible extensions. In particular, for such a graph H and a given instance (G, p) of EXTH, the preassignment $p : X \to V(H)$ is not extendible to H if and only if there exists a *certificate of nonextendability* in the form of a tree with at most k preassigned leaves (where k is a constant depending on H). The existence of such algorithmically well-behaved NO-certificates yields that EXTH is polynomial.

We begin with the concept of a conflict.

DEFINITION 3.3. *Let G and H be graphs. A set $X \subseteq V(G)$ with a preassignment $p : X \to V(H)$ which is not extendible, such that the restriction of p to any proper subset X' of X is extendible, is called a* conflict *in G with respect to H. The* size *of the conflict is $|X|$. A graph H has* strict extension-width *k if every conflict (in any graph G with respect to H) has size at most k.*

Feder and Vardi [13] give several equivalent descriptions of graphs that admit a NUF_k. In particular, the following connection between strict extension-width and near-unanimity is presented.

THEOREM 3.4 (Feder and Vardi [13]). *A graph H admits a* NUF_k *if and only if it has strict extension-width $k-1$.*

A graph has *bounded strict extension-width* if it has strict extension-width k for some integer k. Thus the theorem implies that NUF is precisely the class of graphs with bounded strict extension-width. See [17] for more details on this connection.

Reflexive graphs with a NUF_3 have been most carefully investigated. (Irreflexive graphs with a NUF_3 are characterized in [2].) The class of reflexive graphs with a NUF_3 is known to be the smallest variety containing all reflexive paths [28]. It is also known to be precisely the class of all reflexive graphs H such that H is a retract of any G of which it is an isometric subgraph [19, 28]. What this means in the language of extensions and conflicts is the following. Given a graph G containing H as a subgraph, let $p : X \to V(H)$ be a preassignment where $X = V(H)$ and p is the identity function. Note that in this context (G, p) is naturally viewed as an instance of the retraction problem for H. Either p is extendible (in the case that H is isometric) or there is a conflict of size two (otherwise). Such a conflict yields a *certificate of nonextendability.* In particular, the certificate is a path P with end vertices u and v preassigned as $p'(u) = a, p'(v) = b$, where a, b are vertices of $X = V(H)$. By taking P so that $d_P(u, v) = d_G(a, b) < d_H(a, b)$, we see that p' is extendible in G but $p \circ p'$ is not extendible in H. Hence, p is not extendible to a homomorphism of G to H; i.e., H is not a retract of G. (For a description of NUF_k, $k \geq 3$, as a variety generated by some starting set of building blocks, see [12]. See also [19] and [4].)

We have just observed that graphs in NUF_3 have certificates of nonextendability in the form of a path whose end points have been preassigned. We now extend this concept to larger certificates.

DEFINITION 3.5. *A reflexive graph H has* extension-width one *if for any G and any preassignment $p : X \to V(H)$, where $X \subseteq V(G)$, either p is extendible or there exist a tree T and a set of vertices $X' \subseteq V(T)$ preassigned by a mapping $p' : X' \to X$ such that p' is extendible in G but $p \circ p'$ is not extendible in H. The tree T together with the preassignment p' is called a* tree-certificate. *Furthermore, the tree T is* minimal *if the preassigned vertices are precisely the leaves of T and they form a conflict (in H with respect to T).*

Thus, by the comments above the reflexive graphs with a NUF_3 are precisely the reflexive graphs of extension-width one, where the (minimal) tree-certificates can be chosen to be paths.

In [18] the concept of width one is defined as *tree duality.* A graph H has tree duality if for all G either $G \to H$ or there exists a tree T such that $T \to G$ and $T \not\to H$. The notion of extension-width defined here differs slightly from that in [13]; it has been adapted to extensions (and implicitly to retractions). Thus our condition is that either a preassignment p extends to a homomorphism of G to H or there is a tree T and a preassignment p' from T to the preassigned vertices of G (under p) such that p' extends to a homomorphism of T to G but $p \circ p'$ does not extend to a homomorphism of T to H. Clearly, the existence of a tree-certificate demonstrates that p is not extendible. The proposition below shows that each tree-certificate contains a minimal tree-certificate.

PROPOSITION 3.6. *Let (G, p) be an instance of EXTH where $X \subseteq V(G)$ is the set of preassigned vertices. Suppose T is a tree and $p' : X' \to X$ is a preassignment, where $X' \subseteq V(T)$, such that p' is extendible (in G) but $p \circ p'$ is not extendible (in H). Then T contains, as a subtree, a minimal tree-certificate.*

Proof. Let T' be a minimal subtree of T with respect to nonextendibility in H, and let $X'' \subseteq X'$ be the set of preassigned vertices of in T'. We first claim that each vertex in X'' is a leaf. Suppose to the contrary that some vertex $v \in X''$ is not a leaf. Let $T_1, \ldots, T_k$ be the subtrees of $T' - v$. Consider $T_i + v$. By the minimality of T', $T_i + v$ is extendible in H. Moreover, since v is preassigned the same value in $T_i + v$ for all i, we obtained that the preassignment X'' of T' is extendible in H, which is a contradiction.

We claim that all the leaves in T' are preassigned. Suppose to the contrary that some leaf $v \in T'$ is not preassigned. By minimality, the preassignment from $T' - v$ to $V(H)$ is extendible. In particular, the parent of v receives an image. Since H is reflexive, v can receive the same image. Thus this preassignment of X'' to H is extendible, which is a contradiction.

Therefore, T' is a tree whose leaves are preassigned and the leaves form a conflict. That is, T' together with the preassignment is a minimal tree-certificate. ∎

Tree-certificates are used implicitly in [9] but are first formally defined and studied in [27]. Finally, we remark that K_2 is an example of a graph with width two. That is, for the HOMK_2 problem, i.e., testing if a graph is bipartite, the NO-certificates are odd cycles, i.e., partial two-trees.

A special type of tree-certificate is introduced in [19]. A *hole* in a reflexive graph H is a set Z of vertices with a mapping $\delta : Z \to \{0, 1, 2, \ldots\}$ such that no h in H has the distances $d_H(h, x) \leq \delta(x)$ for all $x \in Z$, but for each proper subset Z' of Z, there is an h satisfying these inequalities for all x in Z'. A *k-hole* is a hole in which the set Z has exactly k vertices. A *unit hole* is a hole in which δ is the constant mapping with range $\{1\}$. If $Z = \{x_1, x_2, \ldots, x_k\}$, we may refer to the hole as a $(\delta(x_1), \delta(x_2), \ldots, \delta(x_k))$ k-hole.

Equivalently, a hole in H is a tree T with exactly one branch vertex (vertex of degree greater than two), together with a precoloring $p : X \to V(H)$, where X is the set of leaves of T. The length of the path from the branch vertex to the leaf x is precisely $\delta(x)$. Further, (X, p) is a conflict in T with respect to H; i.e., p is not extendible in H, but any proper subtree of T, with the same precoloring (restricted to the subtree), is extendible in H.

From the above discussion, we obtain the following corollaries of Theorem 3.4.

COROLLARY 3.7. *A reflexive graph that admits a minimal tree-certificate with k preassigned leaves cannot have a* NUF$_k$.

Proof. The minimal tree-certificate contains a conflict with k vertices, showing the strict extension-width is at least k. ∎

COROLLARY 3.8. *A reflexive graph with a k-hole cannot have a* NUF$_k$.

It turns out that the reflexive graphs with a NUF$_3$ are also characterized as the class of reflexive graphs which do not have holes [19, 28]. Thus a reflexive chordal graph which is not in NUF$_3$ must have a hole. We shall elaborate on this fact in section 4. Specifically, such graphs must have a unit hole of a particular kind.

We now return our attention to improving the bounds on the strict extension-width of reflexive chordal graphs. In Figure 1 is an algorithm from [9]. This algorithm solves the *connected list homomorphism problem*, denoted by the CLIST-HOMH problem, where H is a reflexive chordal graph and the lists of any instance induce connected subgraphs of H. The proof of correctness for the algorithm follows from properties of the perfect elimination ordering of H; see [9]. We present the algorithm with the addition that we explicitly construct a tree-certificate for G in H when G is a NO-instance of the problem. A digraph D is used in the algorithm to retain information

CLIST-HOMH Algorithm [9]

Input: A graph G with lists $L : V(G) \to \mathcal{P}(V(H))$ such that $L(v)$ induces a connected subgraph of H.

Output: A homomorphism $f : G \to H$ such that $f(v) \in L(v)$ for all v, or a tree-certificate T proving $G \not\to H$.

1. Let $h_1, h_2, \ldots, h_n$ be a perfect elimination ordering of H (an ordering such that any two neighbors of h_i among $h_{i+1}, h_{i+2}, \ldots, h_n$ are adjacent).
2. Set $V(D) = \{t_g : g \in V(G)\}$; $E(D) = \emptyset$.
3. For $i = 1$ to n
   - 3.1 Remove h_i from all lists $L(g)$ in which it is not the only member.
   - 3.2 For those g which have $L(g) = \{h_i\}$:
     - 3.2.1 assign $f(g) = h_i$;
     - 3.2.2 for each g' adjacent to g, remove from $L(g')$ all vertices that are not adjacent to h_i; and add the arc $t_{g'}t_g$ to D if some vertex is removed from $L(g')$;
     - 3.2.3 delete g from G.
   - 3.3 If some list $L(x) = \emptyset$, then let T be the subgraph of D consisting of descendants of t_x. Answer NO; return T together with the lists L_T where $L_T(t_g)$ equals the original $L(g)$ (provided as input).
4. Answer YES; return f.

FIG. 1. *An algorithm for CLIST-HOMH, where H is a reflexive chordal graph, and each list induces a connected subgraph of H.*

about which vertices cause the removal of elements from lists (of other vertices). In the case of a NO-instance we prove below, the digraph D contains a tree-certificate (T, p'); thus, H has extension-width one. Also, we provide bounds on the size of the conflict contained in T, thus providing bounds on the strict extension-width of H. That is, we prove the following.

THEOREM 3.9. *Each reflexive chordal graph H has extension-width one and bounded strict extension-width.*

The proof of the theorem appears below after the development of some preliminary results.

LEMMA 3.10. *At any step of the CLIST-HOMH Algorithm in Figure* 1*, the digraph D is acyclic.*

Proof. If $t_{g'}t_g$ is an arc of D, then g is removed from G before g' is removed from G. ∎

LEMMA 3.11. *Suppose the CLIST-HOMH Algorithm in Figure* 1 *answers NO and returns (T, L_T) for some instance (G, L) of CLIST-HOMH. Then for each vertex $t_g \in V(T)$ other than the root, the corresponding vertex $g \in V(G)$ is assigned to some h_i, i.e., $f(g) = h_i$, by the algorithm. Furthermore, this assignment is injective.*

Proof. Suppose t_{g_1} and t_{g_2} are both children of some vertex t_g in T. The arcs $t_g t_{g_1}$ and $t_g t_{g_2}$ in T are created in D after each t_{g_1} and t_{g_2} are assigned an image at step 3.2.2 of the algorithm. Thus, assume that the assignments $f(g_1) = h_i$ and $f(g_2) = h_j$ are made by the algorithm. Without loss of generality we may assume that h_i precedes h_j in the perfect elimination ordering, or $h_i = h_j$. The arc in T from t_g to t_{g_2} was added to T when the assignment $f(g_2) = h_j$ caused some vertex, say, h, to be removed from $L(g)$. In particular, $h_j h \notin E(H)$ and h appears later in the perfect elimination ordering than h_j. Also, $h_i h \in E(H)$; otherwise, h_i causes h to be removed from $L(g)$ before t_{g_2} is assigned h_j, and thus $t_g t_{g_2}$ would not appear in D.

In addition, note that t_g cannot be assigned h_j. Since h_j caused h to be removed from the list, there must be a vertex in the elimination ordering after both h_j and h that will remain in the list for g (after the current round). Recall that the lists are connected. In particular, adjacent vertices receive unique images.

We claim that h_i and h_j are nonadjacent in H. This is immediate from properties of the elimination ordering and the observations $h_ih \in E(H)$, $h_jh \notin E(H)$. The same argument shows there is no path $h_i = u_1, u_2, \ldots, u_m = h_j$ in H where each u_ℓ precedes $u_{\ell+1}$ in the perfect elimination ordering. Indeed, for such a path, $h_jh \notin E(H)$ implies $u_{m-1}h \notin E(H)$, which in turn implies $u_{m-2}h \notin E(H)$. Continuing back to h_i, we conclude $h_ih \notin E(H)$, which is a contradiction.

Finally, there is no vertex h' in the perfect elimination ordering with paths $h = u_1, u_2, \ldots, u_{m_1} = h_i$ and $h = w_1, w_2, \ldots, w_{m_2} = h_j$ such that each u_ℓ precedes $u_{\ell+1}$ and each w_ℓ precedes $w_{\ell+1}$ in the perfect elimination ordering. Suppose to the contrary that such paths exist. Then u_2 and w_2 are common neighbors of h' and must be adjacent. Without loss of generality u_2 precedes w_2 in the elimination ordering. Thus the paths $u_2, u_3, \ldots, u_{m_1} = h_i$ and $u_2, w_2, w_3, \ldots, w_{m_2} = h_j$ are paths with the property above, and the first path has been shortened by one vertex. We thus repeatedly shorten the paths until one has length zero, say, the first path, at which point we have a path from h_i to h_j, which is the case analyzed in the previous paragraph.

This last result about two paths ending at h_i and h_j, respectively, shows that the descendants of t_{g_1} and t_{g_2}, respectively, receive images that are disjoint sets of vertices in H, completing the proof that the function f defined in the algorithm is indeed injective. ∎

We now establish that in the case of a NO-instance, the CLIST-HOMH algorithm does indeed return a tree. Recall that for a list homomorphism problem vertices with lists of size one are called preassigned.

LEMMA 3.12. *If the CLIST-HOMH Algorithm in Figure* 1 *answers NO and returns* (T, L_T)*, then* T *is a tree rooted at* t_x*, where* x *is the vertex whose list becomes empty in step* 3.3*. Moreover, the preassigned vertices are leaves in* T*, and the preassigned images form an independent set in* H*.*

Proof. Every vertex in T other than the root t_x has in-degree one. Suppose to the contrary that some vertex t has predecessors t_1 and t_2. Using the proof of Lemma 3.11, one can easily show that the descendants of t_1 and t_2 are disjoint sets and thus derive a contradiction.

To see that preassigned vertices in T are leaves, observe that internal vertices of T are assigned a value from their list and have something removed from their list by a child. Thus, the list of an internal vertex cannot be a singleton.

Suppose t_{g_1} and t_{g_2} are leaves of T, neither of which is the root. Then they have a common ancestor, say, t_g, such that the (t_g, t_{g_1})-path and the (t_g, t_{g_2})-path in T are internally disjoint. Let $t_{g'_1}$ and $t_{g'_2}$ be the first vertex after t_g on the two paths, respectively, with $t_{g_i} = t_{g'_i}$ possible. Without loss of generality, $f(g_1)$ precedes $f(g_2)$ in the perfect elimination ordering. If $f(g_1)f(g_2) \in E(H)$, then there are paths in H from $f(g_1)$ to $f(g'_1)$ and $f(g_1)$ to $f(g'_2)$, contrary to the proof of Lemma 3.11.

Finally, suppose that the root t' is a preassigned vertex. Then the unique child of the root, say, t_g, causes the single element from $L(t')$, say, h', to be removed when the algorithm makes the assignment $f(g) = h$. Consider some other leaf, say, t_{g_1}, in the tree, including the possibility that t_g is a leaf. Suppose further that the assignment $f(g_1) = h_1$ is made by the algorithm. Then there is a path from h_1 to h in H whose vertices are in increasing order with respect to the perfect elimination ordering. Each vertex on this path is nonadjacent to h'. In particular, $h_1h' \notin E(H)$. Therefore, the set of all leaves is independent. ∎

COROLLARY 3.13. *Suppose* (G, L) *is an instance of EXTH where* $X \subseteq V(G)$ *is the set of preassigned vertices. Further, suppose the CLIST-HOMH Algorithm in Figure* 1 *answers NO and returns* (T, L_T). *Then* T *contains as a subtree a minimal tree-certificate.*

Proof. Suppose (T, L_T) is returned by the CLIST-HOMH Algorithm. Because the lists L_T come from L, each is either a singleton or the entire set $V(H)$. Hence (T, L_T) is also an instance of EXTH, in particular, a NO-instance. Note the mapping $t_g \mapsto g$ is an embedding of T in G. Thus, T is extendible in G but not in H. By Proposition 3.6, T contains as a subtree a minimal tree-certificate. ∎

Proof of Theorem 3.9. Let (G, L) be a NO-instance of EXTH. We will establish that there is a tree-certificate for G with respect to H. Moreover, this tree contains a conflict of size at most $|V(H)|$.

If H is not connected, we consider two cases. First, if some component of G has preassigned vertices in different components of H, then a path between two such vertices is a tree-certificate containing a conflict of size two. On the other hand, if for each component of G its preassigned vertices appear in the same component of H, then (G, L) is a NO-instance if and only if some component of G is a NO-instance for some component of H. Thus, we may restrict our attention to connected graphs G and H. Hence, any instance of EXTH can be viewed as an instance of CLIST-HOMH where each list is a singleton or $V(H)$, and since H is connected, we can apply the algorithm in Figure 1. Given that (G, L) is a NO-instance, the algorithm returns a tree-certificate by Corollary 3.13. Furthermore, this tree has at most $n = |V(H)|$ leaves by Lemma 3.11. The former statement shows that H has extension-width one; the latter statement shows that H has strict extension-width n. ∎

Theorem 3.9 shows that each reflexive chordal graph with n vertices belongs to NUF$_{n+1}$. We now improve the bound.

THEOREM 3.14. *Let H be a reflexive chordal graph with n vertices and maximum clique-size* $\omega \geq 3$. *Then H admits a* NUF$_k$ *with* $k \leq \min\{n - \omega + 1, n - n/(\omega - 1) + 1\}$. *In particular,* $k \leq n - \sqrt{n} + 1$.

Proof. First observe that $n - \sqrt{n} \geq \min\{n - n/(\omega - 1), n - \omega\}$. Hence it suffices to prove that the strict extension-width of H is less than or equal to both $n - n/(\omega - 1)$ and $n - \omega$.

Suppose G is a NO-instance of EXTH. By Corollary 3.13, there is a tree-certificate for G. In particular, there is a tree T with lists L from $V(H)$ that is nonextendible in H. Moreover, the preassigned vertices of T are precisely the leaves of T, and they form a conflict.

Suppose s vertices of T are preassigned values in H. These s vertices can only be leaves of T, including possibly the root t of T, if the root has only one child. Also, by Lemma 3.12, the preassigned values form an independent set in H. Thus, a clique with ω vertices in H contains at most one of the s preassigned images; therefore, the remaining vertices in the clique are not among the s vertices, showing that the strict extension-width is at most $s \leq n - \omega + 1$. This establishes that H admits a NUF$_k$ for $k \leq n - \omega + 2$. In the case that $s = n - \omega + 1$, we note that H must be a split graph with s independent vertices and a clique of size $n - s$. However, such a chordal graph has an intersection representation with $s - 1$ leaves. Thus the leafage of H is at most $s - 1$, and we conclude from Theorem 3.2 that H admits a NUF$_k$ where $k \leq n - \omega + 1$.

Let r be the number of children of t, the root of T. The r children eliminate at least r values from the list for t. If $r \neq 1$, then each of these r vertices must remove at least one value from $L(t)$ which is not among the preassigned images for the leaves of T. Suppose to the contrary that this is not the case. Let t_g be a child of t that removes only values from $L(t)$ that come from the set of preassigned images. Then we can remove the subtree rooted at t_g from T and obtain a smaller tree-certificate, contrary to our assumption that T is minimal. Indeed, each preassigned image, say, h_i, will be removed from the list of all vertices in T at iteration i of the for-loop in the algorithm. Hence, the values that t_g removes from $L(v)$ will all be removed by the algorithm anyway, without the subtree rooted at t_g being present. (Observe that all of the preassigned vertices form an independent set, but the lists are connected. Hence $L(v)$ must contain a vertex that appears later in the perfect elimination ordering than all of the preassigned vertices. Thus v will not be assigned a preassigned image, and they will indeed all be removed from $L(v)$.)

Let i be the number of internal vertices of T (other than the root). Each of these internal vertices is assigned an image which again is not among the preassigned images for the leaves of T. There are l remaining vertices in H that are not one of the $s+i$ assigned images nor one of the r images removed from $L(t)$. Define $p = r + l + i$, unless $r = 1$, in which case $p = l + i$. Thus, $n - s = p$. We will establish the result by showing that $p \geq n/(\omega - 1)$.

Let d be the maximum number of children of a vertex in T other than the root. The number of vertices in H assigned as images to vertices in T other than the root is at most $r + id$, since the root has r children and each of the i internal vertices has at most d children. The r eliminated values from the list for the root t, and the l additional elements counted above, are the only other vertices of H, giving $n \leq 2r + l + id$.

Note finally that $\omega \geq d+1$. If $d = 1$, the claim is trivial. Suppose $d \geq 2$. Consider the d neighbors of an internal vertex t_g (other than the root) of T. Call these neighbors $t_{g_1}, t_{g_2}, \ldots, t_{g_d}$. Further, suppose the algorithm makes the assignments $f(g) = h$ and $f(g_i) = h_i$ for $i = 1, 2, \ldots, d$, and suppose these assignments occur in the order $i = 1, 2, \ldots, d$. The d children remove d vertices from $L(t_g)$. In particular, say that the assignment $f(g_i) = h_i$ causes x_i to be removed from $L(t_g)$. Each of $x_2, x_3, \ldots, x_d$ must appear later in the perfect elimination ordering (by the construction of T) than h_1. Also, each x_i for $2 \leq i \leq d$ must be adjacent to h_1; otherwise, h_1 would have caused x_i to be removed from $L(t_g)$. Hence h_1 is adjacent to h and each of x_i for $2 \leq i \leq d$. Since h_1 precedes the other vertices in the perfect elimination ordering, $h_1, h, x_2, \ldots, x_d$ form a clique of size $d + 1$.

We claim $p(\omega - 1) = (p - i)(\omega - 1) + i(\omega - 1) \geq 2r + l + id \geq n$. To see this note that $\omega - 1 \geq d$, and by hypothesis $\omega - 1 \geq 2$. If $r \neq 1$, then $p - i = r + l$, and the claim follows. When $r = 1$ we have $p - i = l$, giving $l(\omega - 1) \geq 2 + l$ (and the claim), except for $l = 1$ and $\omega = 3$, or $l = 0$. First, note that $l \geq 1$; otherwise, the root t and its only child in T would map to the last two vertices in the perfect elimination ordering, which are adjacent.

Hence, assume that $l = 1$ and $\omega = 3$. If some internal vertex of T other than the root has $d' \leq d$ children, then we claim that $n \leq 2r + l + id - (d - d')$. In this final case the difference of one is accounted for unless $d = d' = 2$. This is not possible since the child of the root t would have two children that remove two values from its list, giving $l \geq 2$, contrary to $l = 1$. Therefore, $p(\omega - 1) \geq n$, or $p \geq n/(\omega - 1)$, completing the proof. ∎

4. Reflexive graphs without NUF.

4.1. Graphs without NUF_k for some k. Having established in the previous sections that each reflexive chordal graph belongs to NUF_k for some k, we now turn our attention to providing lower bounds on the arity of the NUF.

We first show that reflexive chordal graphs without NUF_3 must have a unit hole of a particularly simple kind.

DEFINITION 4.1. *The reflexive chordal graph R_k is formed by constructing a clique on vertices $y_1, \ldots, y_k$ and adding independent vertices $x_1, \ldots, x_k$ and edges $x_i y_j$ for $i \neq j$.*

THEOREM 4.2. *Let H be a reflexive chordal graph which is not in* NUF_3. *There exists $k \geq 3$ such that R_k is an induced subgraph of H. Furthermore, the set $X = \{x_1, x_2, \ldots, x_k\} \subseteq V(R_k)$ is the vertex set of a $(1, 1, \ldots, 1)$ k-hole in H.*

Proof. It suffices to show that H does indeed contain an induced copy of R_k with no vertex adjacent to all the x_i's. Clearly, such a copy of R_k is a $(1, 1, \ldots, 1)$ k-hole in H.

Since $H \notin \mathrm{NUF}_3$, there exist a graph G and a preassignment $p : X \to V(H)$, where $X \subseteq V(G)$, $|X| \geq 3$, and X is a conflict. As described in the proof of Theorem 3.9, we may assume that G and H are connected. (A conflict must belong to a single component of G, and conflicts receiving their images in two components of H have size two.) Hence, by Corollary 3.13, there is a tree-certificate (T, p') for G with respect to H. In particular, $p' : X' \to X$, where X' is the set of leaves of T. Furthermore, $|X'| = |X|$. Suppose to the contrary that $|X'| < |X|$. Since X is a conflict, $p'(X')$ is extendible in H (as it is a proper subset of X). This contradicts the fact that (T, p') is a tree-certificate.

Hence, T is a tree, with at least three leaves which form a conflict with respect to H under the preassignment $p \circ p'$. Without loss of generality, assume T is the smallest tree whose leaves form a conflict of size $|X'|$.

We consider two cases, depending on the degree in T of the root t.

Case 1. $\deg_T(t) = r \geq 3$. Let the children of t be $t_1, t_2, \ldots, t_r$. In the execution of the CLIST-HOMH Algorithm in Figure 1 these are mapped to vertices $x_1, x_2, \ldots, x_r$ of H, respectively. By (the proof of) Lemma 3.12, $X = \{x_1, x_2, \ldots, x_r\}$ is an independent set. Since t is the root of T (meaning that it arose from the first vertex of G whose list became empty), the vertices $x_1, x_2, \ldots, x_r$ have no common neighbor in H. By the minimality of T, for $j = 1, 2, \ldots, r$ the vertices in $X - \{x_j\}$ have a common neighbor y_j in H.

By the minimality of T, for $j \neq i$ the vertex y_i occurs after x_j in the perfect elimination ordering of $V(H)$. Suppose to the contrary that some child of the root, say, x_i, removes a set of images, say, Y_i, from the list of the root, all of which occur before x_j in the perfect elimination ordering. Then we may remove x_i and all of its descendants from T. The algorithm will remove all elements of Y_i from the list of the roots in Step 3.1 before the assignment $f(t_j) = x_j$ is made. Hence, there is no need to include x_i in the tree, contrary to our assumption of minimality. In particular (since $r \geq 3$), any two of $y_1, y_2, \ldots, y_r$ have a common neighbor x_j occurring before them in the perfect elimination ordering. Therefore, $\{y_1, y_2, \ldots, y_r\}$ induces a clique in H, and $\{x_1, x_2, \ldots, x_r\} \cup \{y_1, y_2, \ldots, y_r\}$ induces a copy of R_r, with no vertex in H adjacent to all the x_i's.

Case 2. $\deg_T(t) = r \leq 2$. Since T has $s \geq 3$ leaves, at least one child of the root, say, t_{g_1}, is the root of a subtree of T consisting of a path from t_{g_1} to a vertex t' with at least two children (where $t_{g_1} = t'$ is possible). Let the children of t' be $t_2, \ldots, t_k$ ($k \geq 3$), and assume these vertices have been mapped to $x_2, \ldots, x_k$, respectively, by the algorithm.

If $r = 2$, then let t_{g_2} be the other child of t. Further, let h be the image assigned to t_{g_2} in the algorithm, and define v to be the vertex $v = t_{g_2}$. On the other hand, if $r = 1$, then t is a leaf, and by Corollary 3.13 the original list for t, $L(t)$, must be a singleton, say, $\{h\}$. In this case, define the vertex v to be $v = t$.

Consider the $k-1$ assigned images $x_2, \ldots, x_k$. A vertex x_1 will be defined below. Let y_1 be the value in H assigned to t' by the algorithm. As before, for each i, the $k-2$ vertices x_j, $j \neq i$, have a common neighbor y_i following y_1 in the perfect elimination ordering, with y_i not adjacent to x_i. Hence, for each $i = 2, \ldots, k$, y_1 and y_i have a common neighbor, $x_j, j \neq i$, preceding them in the perfect elimination ordering. Thus, the y_i's, $i = 1, 2, \ldots, k$, form a clique.

Let $d = d_H(y_1, h)$ be the distance in H from y_1 to h. Since T does not map to H, $d_T(t', v) < d_H(y_1, h)$. Let $d' = d_T(t', v)$. Furthermore, for $i \neq 1$, the distance $d_i = d_H(y_i, h)$ must satisfy the inequality $d - 1 \leq d_i \leq d'$. To see the inequality, note that y_1 is adjacent to y_i; hence, $d - 1 \leq d_i$. Also, by the minimality of T, $T - \{x_i\}$ maps to H with t' mapping to y_i. Hence, $d_i \leq d'$.

Consider a tree T' (whose leaves are vertices in H) consisting of a vertex y^* adjacent to each y_i for $i \geq 2$ together with a path of length $d-2$ from y^* to h. If T' does not map to H, then, since T' has distances between its leaves at least their distance in H, the tree T' is another example of a tree like T, but smaller, which is a contradiction. Hence, T' maps to H, and, in particular, y^* maps to a vertex x_1. By the distance inequalities, x_1 is not adjacent to y_1, but it is adjacent to all y_i with $i \neq 1$.

Finally, consider x_i with $i \neq 1$. The vertex x_i is distance at least d from h in H. This follows from the algorithm and properties of the perfect elimination ordering. If h comes after y_1 in the perfect elimination ordering, then a path from x_i to h in H of length at most $d-1$ easily yields a path of length at most $d-1$ from y_1 to h in H, which is a contradiction. Similarly, h preceding y_1 also leads to a contradiction. Since x_1 is at distance $d-2$ from h in H, it follows that x_1 is not adjacent to any x_i with $i \neq 1$. Furthermore, no vertex w in H is adjacent to all x_i, since otherwise we could map T to H by mapping t' to w at distance $d - 1 = d'$ from h in H. This completes the proof. ∎

We continue our examination of holes from page 947, where we defined holes and provided a correspondence to trees with one branch point. The above proof shows that for chordal graphs H without NUF$_3$, there is an instance of EXTH whose tree-certificate is a $(1, 1, \ldots, 1)$ hole. Historically, holes have been studied in the (equivalent) language of retractions. We adopt the retraction language for this next result. Given a graph G with a subgraph H, we say a hole in H is *filled* if G contains a vertex g whose distances to the vertices of the hole are within the distance given in the definition of the hole, i.e., $d_G(g, x) \leq \delta(x)$ for each x in the hole. Thus, a filled hole corresponds to a tree-certificate where the tree is a subdivision of a star. On the other hand, we now provide a graph G that does not retract to a reflexive chordal graph H, yet G does not fill a hole in H. In essence, the tree-certificate for G is necessarily more complex than a subdivision of a star.

THEOREM 4.3. *There is a reflexive chordal graph H with a* NUF$_5$ *(but not a* NUF$_4$*) and an instance G of RETH that does not retract to H, yet no vertex of G fills a hole in H.*

Proof. The graph H has ten vertices a_i, b_i, c_i, d_i, e_i for $i = 1, 2$ and edges a_ic_i, a_id_i, b_ic_i, b_ie_i, c_id_i, c_ie_i, d_ie_i, d_1d_2, e_1e_2, d_1e_2, and e_1d_2. The instance G is H together with three vertices x_1, x_2, y and edges x_ia_i, x_ib_i, x_iy. None of x_1, x_2, y fills a hole in H, and yet G does not retract to H. To see that no hole is filled, observe that $d(x_1, v) \geq d(c_1, v)$ for all $v \in V(H)$. Hence x_1 cannot fill a hole in H. A similar

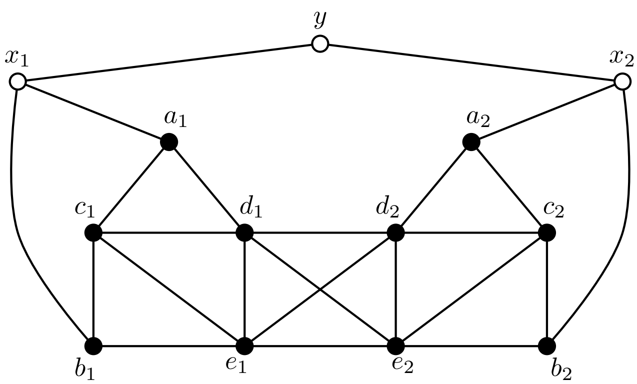

FIG. 2. *A G containing a subgraph H induced by the black vertices. The graph H is in* $\mathrm{NUF}_5 - \mathrm{NUF}_4$. *The graph G is a NO-instance of RETH yet fills no holes in H.*

argument shows that neither x_2 nor y fills a hole. Figure 2 contains a drawing of H (with black vertices) and G (consisting of H plus the three white vertices).

To establish that H admits a NUF_5 and not a NUF_4, we first observe that it is easy to construct an intersection representation for H with leafage 4. Hence, by Theorem 3.2, H admits a NUF_5. Suppose to the contrary that g is a NUF_4 for H. Then $g(a_1, a_1, a_1, c_1) = a_1$ and $g(c_1, b_1, b_1, b_1) = b_1$. Since (a_1, a_1, a_1, c_1) and (c_1, b_1, b_1, b_1) are both adjacent to (d_1, c_1, c_1, e_1), we must have $g(d_1, c_1, c_1, e_1)$ adjacent to both a_1 and b_1. Namely, $g(d_1, c_1, c_1, e_1) = c_1$. However, (d_1, c_1, c_1, e_1) is adjacent to (d_2, d_1, e_1, e_2). Thus, $g(d_2, d_1, e_1, e_2)$ is adjacent to c_1. A similar argument shows that $g(d_2, d_1, e_1, e_2)$ is adjacent to c_2. In particular (a_2, a_2, c_2, a_2) and (b_2, c_2, b_2, b_2) are both adjacent to (c_2, d_2, e_2, c_2). Hence, $g(c_2, d_2, e_2, c_2) = c_2$ and $g(d_2, d_1, e_1, e_2)$ is adjacent to c_2 (as well as c_1), which is a contradiction. ∎

We complete this section by establishing that the bounds in Theorems 3.2 and 3.14 are tight.

THEOREM 4.4. *For each $i \geq 0$ and $\omega \geq 3$, there exists a reflexive chordal graph $LB(\omega, i)$ with $n = 2\omega + i(\omega - 1)$, strict extension-width $s = \omega + i(\omega - 2)$, and leafage s.*

Proof. Fix $\omega \geq 3$.

For the case $i = 0$, $LB(\omega, 0)$ is the chordal graph with vertices $x_1, \ldots, x_\omega$, $y_1, \ldots, y_\omega$, a clique on the y_l, all edges $x_k y_l$ with $k \neq l$, and all loops, i.e., a copy of R_ω as defined above. To see that $LB(\omega, 0)$ has strict extension-width at least $s = \omega$, consider an instance T (a tree defined below) of EXT$LB(\omega, 0)$. The tree T has root t adjacent to ω leaves, each assigned to a unique x_k in $LB(\omega, 0)$. Clearly, the preassignment does not extend to a homomorphism $T \to H$, but if one of the ω leaves in T is deleted, say, the leaf mapped to x_1, then the root t can be mapped to y_1. Thus the x_k's form a conflict of size $s = \omega$. On the other hand, there is an intersection representation of $LB(\omega, 0)$ with s leaves. Hence, by Theorem 3.2, $LB(\omega, 0)$ belongs to NUF_{s+1}. We have $n = 2\omega$, strict extension-width $s = \omega$, and leafage s.

For the case $i = 1$, consider the element x_1 in the preceding construction, with $\omega - 1$ neighbors y_j with $j \neq 1$. To construct $LB(\omega, 1)$, add vertices $z_2, z_3, \ldots, z_\omega$ adjacent to x_1, plus edges $z_k y_l$ for $k \neq l$ and $l \neq 1$. Similarly, replace the leaf in T that is preassigned to x_1 with a vertex t' having the $\omega - 1$ children each preassigned to a unique z_k. Thus n has increased by $\omega - 1$, and the fact that T does not map to $LB(\omega, 1)$ depends on $\omega - 1$ new leaves in T but no longer depends on t' (which is no longer preassigned to x_1). Hence the size of the conflict has increased by at least $\omega - 2$;

however, it is easy to see that the leafage has increased by at most $\omega - 2$. Therefore, s has increased by $\omega - 2$.

Note that the new vertices z_k have $\omega - 1$ neighbors in $LB(\omega, 1)$ as before for the x_k in $LB(\omega, 0)$. We may repeat the process of enlarging each $LB(\omega, i)$ and changing T as follows. In $LB(\omega, i)$ add $\omega - 1$ new vertices, the $z_2, \dots, z_\omega$ above, and join each to some vertex and $\omega - 2$ of its neighbors, as for x_1 above, ensuring that no two of the new vertices have the same neighborhood. In T, select a leaf, say, u, as before, for the preimage of x_1; add $\omega - 1$ leaves each adjacent to u, and map each (injectively) to the new $\omega - 1$ vertices, say, $z_2, \dots, z_\omega$, as above. Thus each time we increase n by $\omega - 1$ and s by $\omega - 2$. This completes the construction. ∎

The bound in Theorem 3.2 is tight.

COROLLARY 4.5. *For each $l \geq 3$, there is a reflexive chordal graph H with $\omega = 3$, and leafage l, such that $H \in \mathrm{NUF}_{l+1} - \mathrm{NUF}_l$.*

Proof. The graph $LB(3, l-3)$ is such a graph. ∎

The bounds in Theorem 3.14 are tight.

COROLLARY 4.6. *For each $\omega \geq 2$ there is a reflexive chordal graph $H \in \mathrm{NUF}_k - \mathrm{NUF}_{k-1}$, where $k = n - \omega + 1$.*

Proof. The graph $LB(\omega, 0)$ is such a graph. ∎

COROLLARY 4.7. *For each $i \geq 0$, there is a reflexive chordal graph $H \in \mathrm{NUF}_k - \mathrm{NUF}_{k-1}$ with $\omega = 3$, where $k = n - n/(\omega - 1) + 1$.*

Proof. The graph $LB(3, i)$ is such a graph. ∎

4.2. Graphs without $\mathbf{NUF}_k$ for any k. We now construct classes of reflexive graphs that do not admit a near-unanimity function of any arity. Given a vertex v and a subgraph S of a graph H, we use the notation $d(v, H)$ to denote the minimum distance from the vertex v to a vertex in the subgraph H, where the minimum is taken over all vertices of H. We are working with reflexive graphs; however, $N(v)$ denotes the vertices other than v that are adjacent to v, i.e., $v \notin N(v)$.

LEMMA 4.8. *Let H be a connected reflexive graph and t, b be two nonadjacent vertices of H. Suppose that for each $x \in N(t)$ there is a complete subgraph $K(x)$, which is either a maximal clique or a single vertex, with the property that $d(x, K(x)) > \max\{d(t, K(x)), d(b, K(x))\}$. Then H does not admit a NUF_k for any $k \geq 3$.*

(The hypotheses are illustrated in Figure 3, where t is the top vertex, b is the bottom vertex, and for each $x \in N(t)$, $K(x)$ is one of the middle four vertices which is not adjacent to x.)

Proof. Suppose to the contrary that g is a NUF_k on H. For each $j = 1, 2, \dots, k$, let $\overline{x}_j$ be a vertex, say, $(x_1, x_2, \dots, x_k)$, of H^k with the property that $x_i = t$ for all $i < j$, $x_i = b$ for all $i > j$, and x_j is an arbitrary vertex of H. Thus $\overline{x}_j = (t, \dots, t, x_j, b, \dots, b)$. Consider any vertex $x \in N(t)$. By hypothesis there exists a complete subgraph $K(x)$ with $d(x, K(x)) > \max\{d(t, K(x)), d(b, K(x))\}$. Thus there exist $y, z \in V(K(x))$ with $d(x, K(x)) > d(t, y)$ and $d(x, K(x)) > d(b, z)$. Hence in H^k the distance between $\overline{x}_j$, and $(y, \dots, y, x_j, z, \dots, z)$ is less than $d(x, K(x))$. For each $v \in K(x)$ the vertex $(y, \dots, y, x_j, z, \dots, z)$ is adjacent to $(v, \dots, v, x_j, v, \dots, v)$. Hence $g(y, \dots, y, x_j, z, \dots, z)$ must be adjacent to v, and as $K(x)$ is either a single vertex or a maximal complete subgraph of H, $g(y, \dots, y, x_j, z, \dots, z)$ must be a vertex of $K(x)$. We have

$$\begin{aligned} d(x, K(x)) &> d(\overline{x}_j, (y, \dots, y, x_j, z, \dots, z)) \\ &\geq d(g(\overline{x}_j), g(y, \dots, y, x_j, z, \dots, z)) \end{aligned}$$

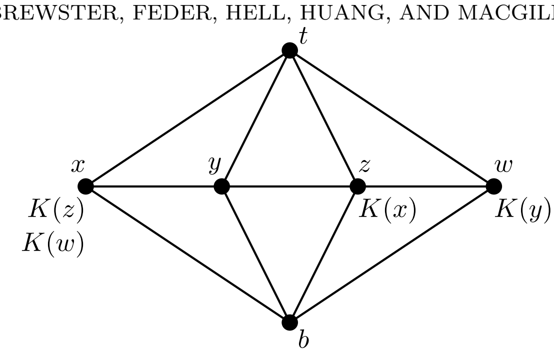

FIG. 3. *A dismantable reflexive graph without a* NUF_k.

and, consequently, $g(\overline{x}_j) \neq x$. Since x was an arbitrary member of $N(t)$, we conclude that $g(\overline{x}_j) \notin N(t)$.

Since H is a connected graph, there is a (t,b)-path $(t =)v_1, v_2, \ldots, v_l(= b)$. Thus, in H^k, we have the following path, say, P:

$$(t, b, b, b, \ldots, b, b) \sim (t, v_{l-1}, b, b, \ldots, b, b) \sim (t, v_{l-2}, b, b, \ldots, b, b) \sim \cdots$$
$$\sim (t, v_2, b, b, \ldots, b, b) \sim (t, t, b, b, \ldots, b, b) \sim (t, t, v_{l-1}, b, \ldots, b, b) \sim \cdots$$
$$\sim (t, t, t, b, \ldots, b, b) \sim (t, t, t, v_{l-1}, \ldots, b, b) \ldots \sim (t, t, t, \ldots, t, b).$$

Since $g(t, b, b, b, \ldots, b) = b$ and $g(t, t, \ldots, t, t, b) = t$, we must have that $g(P)$ is a (b, t)-walk in H. Since t and b are nonadjacent, some interior vertex of P must map to a neighbor of t, contrary to our claim above. □

COROLLARY 4.9. *No reflexive cycle of length at least* 4 *admits a* NUF_k *for any* $k \geq 3$.

Proof. Let $C : v_1v_2 \ldots v_nv_1$ $(n \geq 4)$ be a cycle. We show that C satisfies the assumption of Lemma 4.8: Set $t = v_1$, $b = v_3$, $K_1 = \langle\{v_{\lceil \frac{n-1}{2} \rceil}, v_{\lceil \frac{n}{2} \rceil}\}\rangle$, and $K_2 = \langle\{v_{\lceil \frac{n+3}{2} \rceil}, v_{\lceil \frac{n+4}{2} \rceil}\}\rangle$. Then $N(t) = \{v_2, v_n\}$ and

$$d(v_n, K_1) = \left\lfloor \frac{n}{2} \right\rfloor > \left\lfloor \frac{n}{2} \right\rfloor - 1 = \max\{d(t, K_1), d(b, K_1)\},$$
$$d(v_2, K_2) = \left\lfloor \frac{n}{2} \right\rfloor > \left\lfloor \frac{n}{2} \right\rfloor - 1 = \max\{d(t, K_2), d(b, K_2)\}. \qquad \square$$

We have established that all chordal graphs belong to NUF. Recall that it is proved in [27] that the variety generated by chordal graphs is properly contained in NUF. The corollary below shows that some dismantable graphs do not admit a NUF. It has subsequently been established in [25] that NUF is properly contained in the variety generated by dismantable graphs.

COROLLARY 4.10. *The dismantable reflexive graph in Figure* 3 *does not admit a* NUF_k *for any* $k \geq 3$.

COROLLARY 4.11. *There is a reflexive graph* H *that does not admit a* NUF *of any arity such that every nonidentity retract of* H *admits a* NUF_3.

We complete this section by observing for irreflexive graphs that if H admits a NUF_k, then H must be bipartite. The proof is analogous to Corollary 4.9.

PROPOSITION 4.12. *Let* H *be an irreflexive graph. If* H *is not bipartite, then* H *does not admit a* NUF_k *for any* $k \geq 3$.

5. Conservative NUF. We complete the paper be examining a special type of near-unanimity functions called conservative functions. Recall that $g \in \mathrm{NUF}_k$ is a *conservative near-unanimity function* if $g(x_1, x_2, \ldots, x_k) \in \{x_1, x_2, \ldots, x_k\}$ for all $(x_1, \ldots, x_k)$. Conservative near-unanimity functions have also been called *choice* or *projective* and are of particular interest since if H admits a conservative NUF, then the *list-homomorphism problem* for H is polynomial time solvable.

The computational complexity of LIST-HOMH has been determined for any fixed (general) graph H. In [11] a new class of intersection graphs called *bi-arc* graphs is introduced. The LIST-HOMH problem is polynomial time solvable if H is a bi-arc graph and is NP-complete if H is not a bi-arc graph. This result generalizes the work of [9, 10] since a reflexive graph is a bi-arc graph if and only if it is an interval graph, and an irreflexive graph is a bi-arc graph if and only if it is a complement of a circular arc graph of clique covering number two.

The polynomial time algorithm of [11] is established by showing that every bi-arc graph admits a conservative NUF_3. Under the assumption that P $\neq$ NP, the bi-arc graphs are precisely the graphs with a conservative NUF_3. We now establish this result without the assumption P $\neq$ NP.

Let C be a circle with two specified points p and q. 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 G is a *bi-arc graph* if there is a family of bi-arcs $\{(N_x, S_x) : x \in V(G)\}$ such that, for any not necessarily distinct vertices $x, y \in V(G)$, the following hold:

(i) if x and y are adjacent, neither N_x intersects S_y nor N_y intersects S_x;

(ii) if x and y are not adjacent, both N_x intersects S_y and N_y intersects S_x.

(Note that there cannot be bi-arcs $(N, S), (N', S')$ such that N intersects S' but S does not intersect N' or vice versa.)

We begin by showing that the irreflexive graphs that admit a conservative NUF_3 are precisely the complements of circular arc graphs of clique covering number two.

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. An edge-asteroid which has no edge between $\{u_0, v_0\}$ and $\{v_1, v_2, \ldots, v_{2k}\} \cup V(P_{1,2}) \cup V(P_{2,3}) \cup \ldots \cup V(P_{2k-1,2k})$ is called a *special edge-asteroid.*

We will use the following structural characterization.

THEOREM 5.1 (see [10]). *An irreflexive bipartite graph H is the complement of a circular arc graph if and only if H is chordal bipartite and contains no special edge-asteroids.*

THEOREM 5.2. *Let H be an irreflexive graph. The following are equivalent:*

(a) *H is bipartite and the complement of a circular arc graph;*

(b) *H admits a conservative* NUF_3*;*

(c) *H admits a conservative* NUF_k *for some* $k \geq 3$.

Proof. In [11], it is shown that every complement of a circular arc graph of clique covering number two has a conservative NUF_3. Clearly, if H has a conservative NUF_3, then H has a conservative NUF_k for some k.

To complete the proof, assume to the contrary that H is not the complement of a circular arc graph of clique covering number two, but H does admit a conservative NUF_k, say, g. If H is not bipartite, then by Proposition 4.12 H does not admit a NUF_k for any k. Therefore, H is bipartite.

Suppose H contains an induced (even) cycle, say, C, of length at least six with vertex set $0, 1, 2, \ldots, 2n-1$. In light of Proposition 2.5, g is a conservative NUF$_k$ on the cycle C. Consider the following path in C^k:

$$\begin{aligned}(0, 0, \ldots, 0, 0, 2) &\sim (2n-1, \ldots, 2n-1, 1, 3) \sim (0, \ldots, 0, 2, 2)\\ &\sim (2n-1, \ldots, 2n-1, 1, 3, 3) \sim (0, \ldots, 0, 2, 2, 2)\\ &\sim \cdots \sim (2n-1, 1, 3, \ldots, 3, 3) \sim (0, 2, 2, \ldots, 2, 2).\end{aligned}$$

By near unanimity, the first vertex of the path must map to 0. The vertices of the form $(2n-1, \ldots, 2n-1, 1, 3, \ldots, 3)$ cannot map to 1, since such vertices are distance $n-2$ from $(n+1, \ldots, n+1, n-1, n+1, \ldots n+1)$ and 1 is distance n from $n+1$. Thus the second vertex of the path must map to $2n-1$, the third to 0, the fourth to $2n-1$, etc. By parity, the final vertex maps to 0, which is a contradiction.

Hence H is chordal bipartite. We conclude that H contains a special edge-asteroid. Let the path $P_{0,1}$ be $u_0, p_1, p_2, \ldots, p_t, u_1$. Consider the following path in H^k:

$$\begin{aligned}(u_0, \ldots, u_0, u_0, u_{k+1}) &\sim (v_0, \ldots, v_0, p_1, v_{k+1})\\ &\sim (u_0, \ldots, u_0, p_2, u_{k+1}) \sim (v_0, \ldots, v_0, p_3, v_{k+1})\\ &\sim \cdots \sim (u_0, \ldots, u_0, u_1, u_{k+1}).\end{aligned}$$

Notice that, in this path, all coordinates with the exception of coordinate $k-1$ alternate between u_i and v_i. We will describe this situation by saying that *coordinate* $k-1$ *traverses that path* $P_{0,1}$ *while all other coordinates alternate.*

We claim $g(u_0, \ldots, u_0, u_1, u_{k+1}) = u_0$. Clearly the first vertex of the path is mapped to u_0. Since g is a conservative function and there is no edge between $\{u_{k+1}, v_{k+1}\}$ and $\{v_0, v_1\} \cup P_{0,1}$, we know that no vertex in the above path is mapped to $\{u_{k+1}, v_{k+1}\}$. If $g(u_0, \ldots, u_0, u_1, u_{k+1}) = u_1$, then we look at the path from $(u_0, \ldots, u_0, u_1, u_{k+1})$ to $(u_0, \ldots, u_0, u_1, u_{k+2})$ where the last coordinate traverses the path $P_{k+1,k+2}$ while all other coordinates alternate. Since there is no edge from $\{u_1, v_1\}$ to $\{v_{k+1}, v_{k+2}\} \cup P_{k+1,k+2}$, the image of the entire path must alternate between u_1 and v_1, and thus $g(u_0, \ldots, u_0, u_1, u_{k+2}) = u_1$. Similarly, we consider the path from $(u_0, \ldots, u_0, u_1, u_{k+2})$ to $(u_0, \ldots, u_0, u_2, u_{k+2})$ where coordinate $k-1$ traverses $P_{1,2}$ while all other coordinates alternate. Again there is no edge between $\{u_{k+2}, v_{k+2}\} \cup \{u_0, v_0\}$ and $\{v_1, v_2\} \cup P_{1,2}$. Thus $g(u_0, \ldots, u_0, u_2, u_{k+2}) = u_2$. Continuing in this manner, we can move coordinate k down to u_0 and coordinate $k-1$ up to u_k. We find that $g(u_0, \ldots, u_0, u_k, u_0) = u_k$, which is a contradiction. This proves the claim.

Consider the path from $(u_0, \ldots, u_0, u_1, u_{k+1})$ to $(u_0, \ldots, u_0, u_{k+1}, u_{k+1})$, where the $(k-1)$st coordinate traverses all the paths $P_{1,2}, P_{2,3}, \ldots, P_{k,k+1}$ while the other coordinates alternate. Since there is no edge from $\{u_0, v_0\}$ to any of these paths $P_{i,i+1}$, the image of the entire path must alternate between u_0 and v_0, and thus, $g(u_0, \ldots, u_0, u_{k+1}, u_{k+1}) = u_0$. Continuing in this manner, we can show that $g(u_0, \ldots, u_0, u_{k+1}, \ldots, u_{k+1}) = u_0$ and thus $g(u_0, u_{k+1}, \ldots, u_{k+1}) = u_0$, which is a contradiction. ∎

We conclude by characterizing all graphs that admit a conservative near-unanimity function.

THEOREM 5.3. *Let H be a graph. The following are equivalent.*

(a) *H is a bi-arc graph;*

(b) *H admits a conservative* NUF$_3$*;*

(c) *H admits a conservative* NUF.

Proof. Assume H is a bi-arc graph. In [11], it is shown that H admits a conservative NUF_3. Clearly, if H admits a conservative majority function, then it admits a conservative NUF.

Thus assume H admits a conservative NUF. Let $H^* = H \times K_2$. Note that H^* is irreflexive since K_2 is irreflexive. We have established how to construct a NUF for the product of two graphs given that both factors admit a NUF (see Proposition 2.7). This construction preserves the conservative property. Consequently, H^* is bipartite and the complement of a circular arc graph by Theorem 5.2. Using Proposition 3.1 of [11], we have H is a bi-arc graph. ∎

Note that by Corollary 2.3 and Theorem 5.3, H admits a conservative NUF_3 if and only if H admits a conservative NUF_k for all $k \geq 3$.

It is shown in [11] that the reflexive bi-arc graphs are precisely the interval graphs and the irreflexive bi-arc graphs are precisely the complement of circular arc graphs of clique covering number two. Thus we have the following.

COROLLARY 5.4. *A reflexive graph admits a conservative* NUF *if and only if it is an interval graph and an irreflexive graph admits a conservative* NUF *if and only if it is the complement of a circular arc graph of clique covering number two.*

It is easy to construct a conservative NUF for a reflexive (resp., irreflexive) bi-arc graph using an interval (resp., a circular arc) representation of the graph. In [11], a direct construction of a conservative NUF is given for bi-arc trees. However, a direct construction for general bi-arc graphs still remains elusive.

Acknowledgment. The authors wish to thank the referees for a very careful reading of the manuscript and many helpful suggestions.

REFERENCES

[1] K. A. Baker and A. F. Pixley, *Polynomial interpolation and the Chinese remainder theorem for algebraic systems*, Math. Z., 143 (1975), pp. 165–174.
[2] H. Bandelt, *Graphs with edge-preserving majority functions*, Discrete Math., 103 (1992), pp. 1–5.
[3] H. J. Bandelt, M. Farber, and P. Hell, *Absolute reflexive retracts*, Discrete Appl. Math., 44 (1993), pp. 9–20.
[4] R. C. Brewster and G. MacGillivray, *Building blocks for the variety of absolute retracts*, Discrete Math., 306 (2006), pp. 1758–1764.
[5] A. A. Bulatov, *Tractable conservative constraint satisfaction problems*, in Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (Ottawa, Canada, 2003), IEEE Computer Society, Silver Spring, MD, pp. 321–330.
[6] V. Dalmau and A. Krokhin, *Majority constraints have bounded pathwidth duality*, European J. Combin., to appear.
[7] P. Dukes, H. Emerson, and G. MacGillivray, *Undecidable generalized colouring problems*, J. Combin. Math. Combin. Comput., 26 (1998), pp. 97–112.
[8] T. Feder, *Classification of homomorphisms to oriented cycles and of k-partite satisfiability*, SIAM J. Discrete Math., 14 (2001), pp. 471–480.
[9] T. Feder and P. Hell, *List homomorphisms to reflexive graphs*, J. Combin. Theory Ser. B, 72 (1998), pp. 236–250.
[10] T. Feder, P. Hell, and J. Huang, *List homomorphisms and circular arc graphs*, Combinatorica, 19 (1999), pp. 487–505.
[11] T. Feder, P. Hell, and J. Huang, *Bi-arc graphs and the complexity of list homomorphisms*, J. Graph Theory, 42 (2003), pp. 61–80.
[12] T. Feder, P. Hell, B. Larose, C. Loten, and C. Tardif, *Near Unanimity Functions and Tree Duals*, manuscript, 2008.
[13] 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), pp. 57–104.

[14] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
[15] P. Hell, *Retractions de graphes,* Ph.D. thesis, Université de Montreal, Montreal, Canada, 1972.
[16] P. Hell and J. Nešetřil, *On the complexity of H-colouring*, J. Combin. Theory Ser. B, 48 (1990), pp. 92–110.
[17] P. Hell and J. Nešetřil, *Graphs and Homomorphisms*, Oxford Lecture Series in Mathematics and Its Applications 28, Oxford University Press, Oxford, UK, 2004.
[18] P. Hell, J. Nešetřil, and X. Zhu, *Duality and polynomial testing of tree homomorphisms*, Trans. Amer. Math. Soc., 348 (1996), pp. 1281–1297.
[19] P. Hell and I. Rival, *Absolute retracts and varieties of reflexive graphs*, Canad. J. Math., 39 (1987), pp. 544–567.
[20] A. P. Huhn, *Schwach distributive Verbände*. I, Acta Sci. Math. (Szeged), 33 (1972), pp. 297–305.
[21] E. M. Jawhari, D. Misane, and M. Pouzet, *Retracts: Graphs and ordered sets from the metric point of view*, in Combinatorics and Ordered Sets, Contemp. Math. 57, AMS, Providence, RI, 1986, pp. 175–226.
[22] H. W. Kin and V. Dalmau, *Generalized majority-minority operations are tractable*, in Proceedings of the Twentieth Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society, Washington, DC, 2005, pp. 438–447.
[23] G. Kun, B. Larose, and A. Memartolui, *Private communication*, 2004.
[24] B. Larose, C. Loten, and C. Tardif, *A characterisation of first-order constraint satisfaction problems*, in Proceedings of the 21st IEEE Symposium on Logic in Computer Science, IEEE Computer Society, Washington, DC, 2006, pp. 201–210.
[25] B. Larose, C. Loten, and L. Zádori, *A polynomial time algorithm for near-unanimity graphs*, J. Algorithms, 55 (2005), pp. 177–191.
[26] I. Lin, T. McKee, and D. West, *The leafage of a chordal graph*, Discuss. Math. Graph Theory, 18 (1998), pp. 23–48.
[27] C. Loten, *Retractions of Chordal and Related Graphs*, Ph.D. thesis, Simon Fraser University, Burnaby, BC, Canada, 2003.
[28] R. Nowakowski and I. Rival, *The smallest graph variety containing all paths*, Discrete Math., 43 (1983), pp. 223–234.
[29] R. Nowakowski and P. Winkler, *Vertex-to-vertex pursuit in a graph*, Discrete Math., 43 (1983), pp. 235–239.
[30] E. Pesch, *Retracts of Graphs,* Mathematical Systems in Economics 110, Athenäum, Frankfurt am Main, 1988.
[31] R. W. Quackenbush, I. Rival, and I. Rosenberg, *Clones, order varieties, near unanimity functions and holes*, Order, 7 (1990), pp. 239–247.