

11-26-2010

Building Graphs from Colored Trees

Rachel M. Esselstein

California State University, Monterey Bay

Peter Winkler

Dartmouth College

Follow this and additional works at: <https://digitalcommons.dartmouth.edu/facoa>



Part of the [Discrete Mathematics and Combinatorics Commons](#)

Recommended Citation

Esselstein, Rachel M. and Winkler, Peter, "Building Graphs from Colored Trees" (2010). *Open Dartmouth: Faculty Open Access Articles*. 2322.

<https://digitalcommons.dartmouth.edu/facoa/2322>

This Article is brought to you for free and open access by Dartmouth Digital Commons. It has been accepted for inclusion in Open Dartmouth: Faculty Open Access Articles by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.

Building Graphs from Colored Trees

Rachel M. Esselstein

CSUMB Department of Mathematics and Statistics
100 Campus Center Dr. Building 53
Seaside, CA 93955, U.S.A.

`resselstein@csumb.edu`

Peter Winkler*

Department of Mathematics, Kemeny Hall
Dartmouth College
Hanover, NH 03755, U.S.A.

`peter.winkler@dartmouth.edu`

Submitted: Sep 7, 2009; Accepted: Oct 7, 2010; Published: Nov 26, 2010

Mathematics Subject Classification: 05C85

Abstract

We will explore the computational complexity of satisfying certain sets of neighborhood conditions in graphs with various properties. More precisely, fix a radius ρ and let $N(G)$ be the set of isomorphism classes of ρ -neighborhoods of vertices of G where G is a graph whose vertices are colored (not necessarily properly) by colors from a fixed finite palette. The *root* of the neighborhood will be the unique vertex at the “center” of the graph. Given a set \mathcal{S} of colored graphs with a unique root, when is there a graph G with $N(G) = \mathcal{S}$? Or $N(G) \subset \mathcal{S}$? What if G is forced to be infinite, or connected, or both?

If the neighborhoods are unrestricted, all these problems are recursively unsolvable; this follows from the work of Bulitko [Graphs with prescribed environments of the vertices. *Trudy Mat. Inst. Steklov.*, 133:78–94, 274, 1973]. In contrast, when the neighborhoods are cycle free, all the problems are in the class P. Surprisingly, if G is required to be a regular (and thus infinite) tree, we show the realization problem is NP-complete (for degree 3 and higher); whereas, if G is allowed to be any finite graph, the realization problem is in P.

*Research supported in part by NSF Grant DMS-0600876

1 Introduction

The notion of a neighborhood will be used throughout this paper in two different ways. Formally, the *neighborhood of radius ρ of a vertex v in graph G* is the subgraph given by the set of all vertices reachable from v in G by a path of length $\leq \rho$ and the corresponding edges from G . We say that v is the *center* of this neighborhood. In this paper we will often refer to neighborhoods of radius ρ without referring to a graph G . In this case, we are referring to a different notion of a neighborhood as a finite graph with a distinguished vertex called the *center* such that every vertex in the graph is of distance $\leq \rho$ from the center. This abuse of notation derives from our goal of trying to construct graphs such that every vertex in the graph has its neighborhood in a given set. We will sometimes refer to the *ρ -neighborhood set* of a graph G by which we mean the set of all isomorphism classes of ρ -neighborhoods of the graph G . We will denote the isomorphism class of a ρ -neighborhood of a vertex v by $N_\rho(v)$.

Given a set \mathcal{S} of neighborhoods of radius ρ , we want to determine whether there is a colored graph such that every vertex in the graph is the center of one of the neighborhoods in \mathcal{S} . We define the following:

1. The set \mathcal{S} is *sufficient* if there is a colored connected graph G such that every ρ -neighborhood in G belongs to \mathcal{S} . In this case, the graph G is said to *satisfy* \mathcal{S} .
2. The set \mathcal{S} is *consistent* if for every ρ -neighborhood $N \in \mathcal{S}$, there is a colored connected graph G which uses only neighborhoods from \mathcal{S} and which exhibits the ρ -neighborhood N .
3. The set \mathcal{S} is *realizable* if there is a colored connected graph G such that every ρ -neighborhood in G belongs to \mathcal{S} and every neighborhood from \mathcal{S} appears in G . The graph G is said to *realize* \mathcal{S} .

We would like to know the complexity of determining whether a given set \mathcal{S} of ρ -neighborhoods with vertices colored by c colors is sufficient, consistent, or realizable via a connected graph. Notice that if we allowed the graphs to be disconnected, then realizability and consistency would be the same concept. Also, a set \mathcal{S} is sufficient (respectively, consistent) via a connected graph if and only if it is also sufficient (respectively, consistent) via a disconnected graph. Thus, we will only be concerned with building connected graphs.

The following results were motivated by the problem of determining expressibility in monadic second-order logic, in which all elements of a structure are “colored.” [4] In order to prove that a property is expressible in monadic second-order logic, one must construct a partial isomorphism between two colored structures, one which models the property and one which does not. Thus, the problem of preserving local properties of the structure becomes quite important. For more on this, see [1], [3] or [7]. The neighborhoods in this article are the local properties being preserved and the larger graph we build from them is a model of the desired theory.

We begin by noting that

Realizability \rightarrow Consistency \rightarrow Sufficiency,

where each arrow represents implication and none of the arrows are reversible. Most of this result is straightforward and thus will be left to the reader, however we will include the proof of the proposition that consistency does not imply realizability.

1.1 Proposition. *There is a set \mathcal{S} of ρ -neighborhoods of degree d such that \mathcal{S} is consistent but not realizable.*

Proof. Let every neighborhood in \mathcal{S} be cycle-free with degree $d = 2$ and radius $\rho = 1$. Denote each neighborhood by a pair $\langle x; X \rangle$ where x is the color of the root and X is the multiset of the colors of the leaves. Note that the order of the leaves does not matter; only the multiplicity of the colors is important. The set of colors we use is $\{1, 2, 3, 4, 5, 6, 2', 3', 4'\}$.

We claim that the set

$$\begin{aligned} \mathcal{S} = \{ & \langle 1; 2, 2' \rangle, \langle 2; 1, 3 \rangle, \langle 3; 2, 4 \rangle, \langle 4; 3, 2 \rangle, \langle 2; 4, 3 \rangle, \\ & \langle 2'; 1, 3' \rangle, \langle 3'; 2', 4' \rangle, \langle 4'; 3', 2' \rangle, \langle 2'; 4', 3' \rangle, \\ & \langle 2; 1, 5 \rangle, \langle 5; 2, 6 \rangle, \langle 6; 5, 2 \rangle, \langle 2; 6, 5 \rangle \}, \end{aligned}$$

is a consistent but not realizable set.

It is not difficult to check that \mathcal{S} is consistent. This is left to the reader.

We will show that \mathcal{S} is not realizable. Without adding extra structure, we can add an orientation to the edges of each of the neighborhoods as follows: $1 \rightarrow 2$, $1 \rightarrow 2'$, $2 \rightarrow 3$, $2' \rightarrow 3'$, $3 \rightarrow 4$, $3' \rightarrow 4'$, $4 \rightarrow 2$, $4' \rightarrow 2'$, $2 \rightarrow 5$, $5 \rightarrow 6$, and $6 \rightarrow 2$. This gives $\langle 1; 2, 2' \rangle$ an out-degree of 2 and all other neighborhoods, an out-degree of 1 and an in-degree of 1. Thus, any graph exhibiting *all* the neighborhoods of \mathcal{S} will have as its root $\langle 1; 2, 2' \rangle$, and then have directed paths always leading away from the root.

It is clear that we need $\langle 1; 2, 2' \rangle$ to appear twice in a graph if both the neighborhoods $\langle 2; 1, 3 \rangle$ and $\langle 2; 1, 5 \rangle$ are to appear. Unfortunately, this is impossible since two instances of $\langle 1; 2, 2' \rangle$ in a connected graph would create a directed path which leads toward one of the copies of $\langle 1; 2, 2' \rangle$ (which, by our construction, cannot happen). It follows that \mathcal{S} is not realizable. \square

This proof illustrates a set of *rotator neighborhoods* in which the colors dictate that every path in a graph built from \mathcal{S} is ultimately periodic. Such neighborhoods will be important in later proofs.

In general, we can not determine recursively whether an arbitrary set of neighborhoods with colored vertices can be realized by a connected graph. This is proven as a special case of [2]. In attempt to improve the computational complexity of this problem, we will look at a more restricted class of neighborhoods; specifically, neighborhoods with no cycles.

A (ρ, d, C) -tree neighborhood is a rooted complete tree of height ρ where every vertex which is not a leaf has fixed degree d . The vertices of each of our (ρ, d, C) -tree

neighborhoods will be colored by a set of C colors. When ρ , d and C are understood, we will simply refer to these as tree neighborhoods. Every neighborhood in the rest of this paper will be a tree neighborhood unless otherwise indicated.

In this paper we are interested in evaluating the computational complexity of the consistency, sufficiency and realizability of sets of (ρ, d, C) -tree neighborhoods. In section 2 we will look at the complexity of determining the existence of an infinite tree realizing a set of tree neighborhoods. In section 3 we will look at the complexity of determining the existence of finite graphs realizing a set of neighborhoods.

2 Realizability via Infinite Trees

In this section we will explore the computational complexity of determining the existence of an infinite tree realized by a set \mathcal{S} of tree neighborhoods. In the case that the degree of the tree neighborhoods is 2, the problem can be shown to be in P. We omit this proof but include the proofs that for neighborhoods of degree 3 and higher the problem is NP-complete as these proofs are more illustrative.

Let \mathcal{S} be a set of tree neighborhoods (with distinguished center) of radius ρ and fixed degree $d \geq 3$. Enumerate the neighborhoods in \mathcal{S} and, for each neighborhood, the edges incident to the center vertex.

The *de Bruijn graph* \mathcal{B} corresponding to the set \mathcal{S} has vertex set $\{N_i : N \in \mathcal{S}, i \in \{1, 2, \dots, d\}\}$. There is a solid black edge between vertices N_i and M_j if and only if the center vertex of N is the color of the j th edge extending the center vertex of M (according to the enumeration of the edges), the center vertex of M is the color of the i th edge extending the center vertex of N , and the $(\rho-1)$ -neighborhood of the center vertex of N (respectively M) is completely contained in M (respectively N). There is a dashed edge in \mathcal{B} between vertices N_i and N_j for every neighborhood N in \mathcal{S} and every $i, j \in \{1, 2, \dots, d\}$.

For example, consider the set of 1-neighborhoods $\mathcal{S} = \{\langle r; s, s, t \rangle, \langle s; t, t, r \rangle, \langle t; r, r, s \rangle\}$ where the edges are enumerated according to the order they appear in the neighborhood notation. The de Bruijn graph corresponding to \mathcal{S} is given in Figure 1.

Notice that the de Bruijn graph corresponding to \mathcal{S} is constructible in polynomial time with respect to n (the number of neighborhoods in \mathcal{S}) and c (the number of colors).

A *walk* in the de Bruijn graph is a path that follows certain rules:

- In the initial step, starting at some vertex N_i , one may travel along any solid edge to any vertex in \mathcal{B} .
- For each subsequent step, one alternates traveling along dashed edges and solid edges.

The length of the walk is measured by the number of solid edges traveled. A walk in \mathcal{B} corresponds to a path through a tree visiting only neighborhoods from \mathcal{S} .

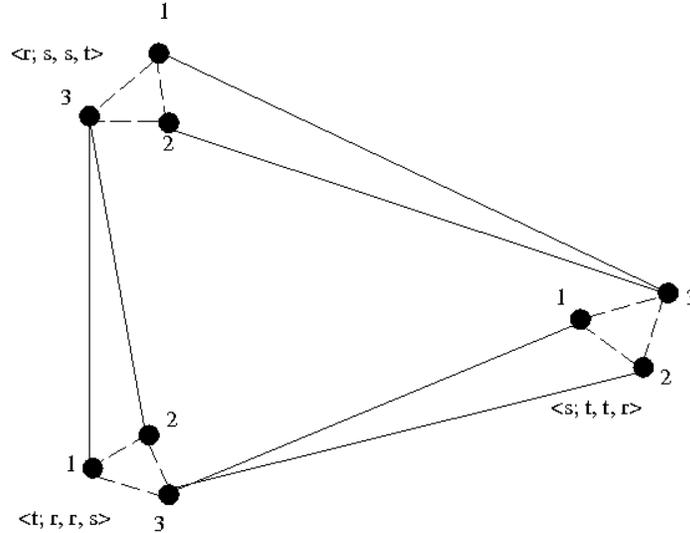


Figure 1: de Bruijn graph on \mathcal{S}

We will be interested in finding the minimal distance (i.e. shortest path) between any pair of neighborhoods in \mathcal{S} . The de Bruijn graph will help us find this bound.

Suppose there is a path from the vertex corresponding to the i th edge of neighborhood N to any of the vertices corresponding to neighborhood M . We claim that if any neighborhood appears more than twice along this path, then the path may be shortened. The only time we would need a neighborhood to appear twice along this path is if the vertex we arrive on is also the vertex we wish to leave from. In this case, we leave the neighborhood along a different edge and then return to the vertex corresponding to the desired edge. Thus, if there are n neighborhoods in \mathcal{S} , we may need to visit every neighborhood except M twice. Therefore, there is a path from N_i to M in \mathcal{B} of length no more than $2n - 1$. Notice that this works for any radius ρ . We now show that if there is an infinite tree realizing \mathcal{S} , then there is a finite tree whose interior vertices realize \mathcal{S} .

2.1 Proposition. *If the set \mathcal{S} (containing n neighborhoods) is realizable by an infinite tree, then there is a finite tree of radius $n(2n - 1 + \rho)$ in which every vertex of degree d has its neighborhood in \mathcal{S} and every neighborhood in \mathcal{S} appears as the neighborhood of at least one of the vertices of degree d .*

Proof. Suppose there is an infinite tree realizing \mathcal{S} . Mark one copy of each neighborhood from \mathcal{S} in the tree. One at a time, connect the center vertex of each marked neighborhood to the previous marked neighborhood via a path through the tree. Prune the rest of the tree to yield a finite, connected tree. The resulting tree might have a diameter greater than $2n - 1 + \rho$. Since each path through the tree represents a walk in the de Bruijn graph, we can use the argument above to shorten the paths between two marked neighborhoods. Assume π is a path of length greater than $2n - 1 + \rho$ connecting two marked neighborhoods such that the only marked neighborhoods in π are the endpoints. We use the argument

above to shorten this path to a new path of length less than or equal to $2n - 1 + \rho$. If we do this for each path with no intermediate marked neighborhoods then the diameter of the resulting graph will be less than or equal to $n(2n - 1 + \rho)$. \square

Determining the existence of an infinite tree realizing \mathcal{S} is in NP since the set \mathcal{S} is realizable if and only if there is a finite tree which exhibits every neighborhood from \mathcal{S} and \mathcal{S} is consistent. We showed in Lemma 2.1 that if there is an infinite tree realizing \mathcal{S} , then there is a finite tree realizing \mathcal{S} . The consistency of \mathcal{S} guarantees that we may extend each of the branches indefinitely to an infinite tree realizing \mathcal{S} . It is easily seen that the consistency of \mathcal{S} is decidable in polynomial time and that a nondeterministic polynomial time machine can guess a finite tree from the list of all possible finite trees of radius $n(2n - 1 + \rho)$ and accept if and only if the interior vertices give a realization of \mathcal{S} . Thus, realizing \mathcal{S} by an infinite tree is in NP.

We will use these results to show that the problem of deciding whether \mathcal{S} is realizable by an infinite tree is NP-complete. Our proof is by a reduction from a variant of 3-SAT. The standard version of 3-SAT allows clauses in which the same literal appears more than once. We call such clauses *repetitive*. For example, 3-SAT allows the repetitive clause $(x \vee x \vee \bar{y})$. For our reduction we would like to ensure that none of our clauses are repetitive. Thus, we would allow such clauses as $(x \vee \bar{x} \vee \bar{y})$ but not the repetitive clause above in which literal x appears twice. It is not difficult to find a reduction of this type of 3-SAT, called STRICT 3-SAT, to the classical 3-SAT.

2.2 Theorem. *Let \mathcal{S} be a set of tree neighborhoods of radius 1 and degree 3. The problem of determining the existence of an infinite tree realizing \mathcal{S} is NP-complete.*

Proof. We showed that this problem is in NP. Now we perform a reduction from STRICT 3-SAT to our realizability problem. Consider an instance of STRICT 3-SAT given by $\Phi = \{c_1, c_2, \dots, c_m\}$ where each c_i is a disjunction of three literals over a set of n variables, V , such that none of the clauses in Φ are repetitive. Φ is satisfied if and only if there is a truth valuation which satisfies at least one literal in each clause c_i .

We want a polynomial time algorithm to construct a set \mathcal{S} of colored neighborhoods ($d = 3, \rho = 1$) such that Φ is satisfied if and only if \mathcal{S} is realizable.

Choose $k \geq 0$ such that the number of variables n in V is $3 \cdot 2^k$ (if necessary, add “dummy variables” which are not used in any of the clauses in Φ to force $n = 3 \cdot 2^k$).

We begin our construction of \mathcal{S} with a collection of neighborhoods called the *core neighborhoods* that must appear exactly once in any tree realizing \mathcal{S} . These are $\langle 0; 1, 1, 1 \rangle$ and, $\langle 1; 0, 2, 2 \rangle, \langle 2; 1, 3, 3 \rangle, \dots, \langle k + 1; k, k + 2, k + 2 \rangle$, where each number $0, 1, 2, \dots, k + 2$ represents a distinct color.

We surround the core with *variable neighborhoods* which consist of

$$\langle k + 2; k + 1, x_0, x_0 \rangle, \text{ and } \langle k + 2; k + 1, x_0, r \rangle$$

for each variable $x \in V$, where x_0 is a color corresponding to the variable x and r is a new color which will be explained later in the construction.

Next, we include a collection of neighborhoods called the *literal neighborhoods*.

If the literal x (respectively \bar{x}) does not appear in clause c_1 , we include

$$\langle x_0; k + 2, x_1, r \rangle, \text{ resp. } \langle x_0; k + 2, \bar{x}_1, r \rangle$$

where x_1 and \bar{x}_1 are new colors. Otherwise, we include new colors g_1 and h_1 and the neighborhoods

$$\langle x_0; k + 2, x_1, g_1 \rangle, \langle g_1; x_1, h_1, r \rangle,$$

respectively

$$\langle x_0; k + 2, \bar{x}_1, g_1 \rangle, \langle g_1; \bar{x}_1, h_1, r \rangle.$$

For each subsequent clause c_i in Φ ($2 \leq i \leq m$), if the literal x does not appear in c_i , we include the neighborhood

$$\langle x_i, x_{i-1}, x_{i+1}, r \rangle.$$

Otherwise, we introduce new colors g_i and h_i and the neighborhoods

$$\langle x_i, x_{i-1}, x_{i+1}, g_i \rangle, \text{ and } \langle g_i; x_i, h_i, r \rangle.$$

Similarly for the literal \bar{x} .

Finally, we include a final layer of literal neighborhoods

$$\langle x_{m+1}, x_m, r, r \rangle, \text{ and } \langle \bar{x}_{m+1}, \bar{x}_m, r, r \rangle.$$

(Note that the literals x_{m+1} and \bar{x}_{m+1} do not correspond to any clauses in Φ .)

Then come the *satisfaction neighborhoods*. We add four more colors a_1, a_2, a_3 and a_4 and for all $1 \leq i \leq m$, include the following neighborhoods:

$$\langle h_i; g_i, a_1, r \rangle, \langle h_i; g_i, a_2, r \rangle, \langle h_i; g_i, a_3, r \rangle, \langle h_i; g_i, a_3, r \rangle,$$

and

$$\langle a_1; h_i, r, r \rangle, \langle a_2; h_i, r, r \rangle, \langle a_3; h_i, r, r \rangle, \langle a_4; h_i, r, r \rangle.$$

Many of the neighborhoods above contain a vertex colored by r . For every neighborhood containing such a vertex we include the set of *rotator neighborhoods*

$$\langle r; *, s, s \rangle, \langle s; r, t, t \rangle, \langle t; s, r, r \rangle, \langle r; t, s, s \rangle,$$

where $*$ is the color of the center vertex of a neighborhood containing a vertex colored by r .

This completes our construction of \mathcal{S} . We now show that Φ is satisfiable if and only if \mathcal{S} is realizable.

Suppose Φ is satisfiable. Then there is a truth assignment t which satisfies Φ . We use this truth assignment to construct the realization of \mathcal{S} in levels.

Begin with the neighborhood $\langle 0; 1, 1, 1 \rangle$ and extend in the canonical way out to the $k + 2$ nd level using the core neighborhoods. Then extend each branch using the variable neighborhoods so that every variable appears on the next level exactly 3 times. Since we chose k such that $V = 3 \cdot 2^k$, this is always possible.

We use the truth valuation t to extend to the next level. For each literal, we extend using two copies of the neighborhood which corresponds to the truth value given by t and only one copy of the other neighborhood. For example, if $t(x) = \mathbf{False}$ then we include two copies of the literal neighborhood $\langle x_0; k + 2, \overline{x_1}, r \rangle$ and exactly one copy of the literal neighborhood $\langle x_0; k + 2, x_1, r \rangle$. We extend the branches containing x_i for $i = 1, \dots, m+1$. By construction, if the literal corresponding to the x_i appears in clause c_i , we will be forced to branch off by g_i and then h_i .

At this stage in the construction of the tree, there are two branches for every literal, given by the truth valuation which satisfies Φ . Since each clause is satisfied by the truth valuation, there is at least one literal in each clause satisfied by it. Thus, there are between 4 and 6 branches in the tree extending to g_i and then h_i for each $1 \leq i \leq m$. So since Φ was satisfiable, we should have at least 4 neighborhoods of the form $\langle h_i; g_i, r, * \rangle$ for each $1 \leq i \leq m$. To ensure finite realizability, we are forced to use each of the satisfaction neighborhoods $\langle h_i; g_i, a_j, r \rangle$ in the tree. We then complete the infinite tree by extending all the branches by rotator neighborhoods indefinitely. Thus we have shown that if Φ is satisfiable, the set \mathcal{S} is realizable.

Now, we assume that \mathcal{S} is realizable and show that Φ is satisfiable. We first make some comments on what a realization of \mathcal{S} must look like.

Without adding any extra structure, we may add an orientation to the edges of the neighborhoods of \mathcal{S} as follows:

$\langle 0; 1, 1, 1 \rangle$ has out-degree 3 and all other neighborhoods of \mathcal{S} have in-degree 1 along the first arm and out-degree 2 along the other two arms. Thus any graph built from neighborhoods of \mathcal{S} can only have paths leading away from $\langle 0; 1, 1, 1 \rangle$. This gives us that $\langle 0; 1, 1, 1 \rangle$ can only appear once in any realization of \mathcal{S} since otherwise the path connecting the two copies of $\langle 0; 1, 1, 1 \rangle$ must lead toward one of them which cannot happen.

Assume G is a realization of \mathcal{S} . We know G must have exactly one copy of $\langle 0; 1, 1, 1 \rangle$. There is only one way to extend each of the branches to complete the first $k + 2$ levels of the tree.

Thus G must have the “canonical core” as illustrated in Figure 2.

At the $k + 3$ rd level there will be exactly enough free arms to exhibit both of the following neighborhoods from \mathcal{S} for each literal x :

$$\langle k + 2; k + 1, x_0, x_0 \rangle, \text{ and } \langle k + 2; k + 1, x_0, r \rangle.$$

Since the color $k + 2$ may only appear in one level of the tree, G must realize each of these neighborhoods exactly once on this level.

Thus we have three free arms extending x_0 for each literal x .

There are only two neighborhoods in \mathcal{S} which can extend x_0 :

$$\langle x_0; k + 2, x_1, r \rangle \text{ and } \langle x_0; k + 2, \overline{x_1}, r \rangle.$$

The color x_0 may only appear in one level so the colors x_1 and $\overline{x_1}$ may only appear in the $k + 4$ th level. Thus for each variable x , one of these neighborhoods must appear once and the other must appear twice in order to realize \mathcal{S} .

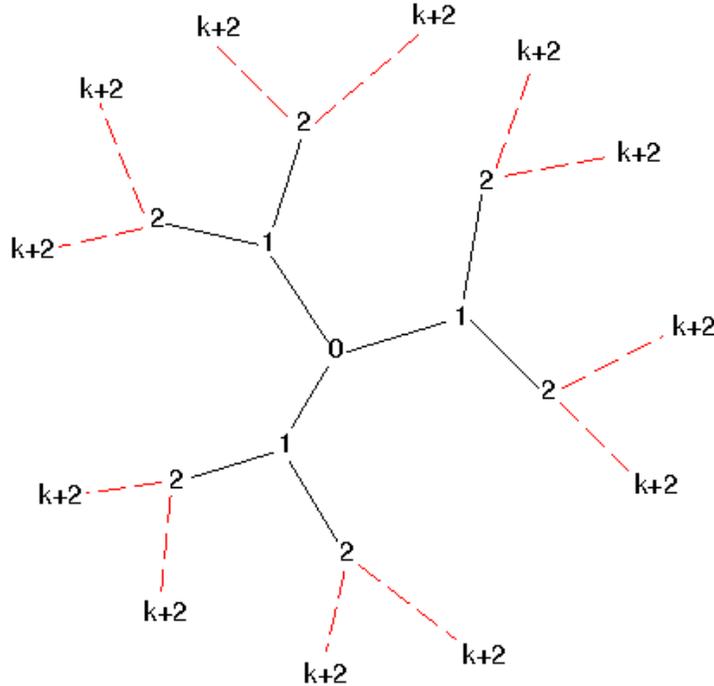


Figure 2: The “canonical core” where the dashed lines denote the possibility of several more levels or possibly none before reaching the $k + 2$ nd level.

From here we canonically extend x_i to x_{i+1} and $\overline{x_i}$ to $\overline{x_{i+1}}$ until $i = m$. If the literal corresponding to the branch appears in clause c_i , we must extend the free arm by g_i and otherwise we must always extend the free arm by r . The color g_i must always be extended by $\langle g_i; x_i, h_i, r \rangle$.

At this stage in the graph, all of the neighborhoods $\langle h_i; g_i, r, * \rangle$ can be extended by any of the a_j for $1 \leq j \leq 4$. However, in order for \mathcal{S} to be realized, G must have at least 4 copies of $\langle h_i; g_i, r, * \rangle$ for every $1 \leq i \leq m$. Each clause c_i has exactly 3 literals and so there is one branch connecting to g_i and h_i for every literal appearing in c_i which is not satisfied by the interpreted truth valuation on the $k + 4$ th level, and two branches connecting to g_i and h_i for every literal appearing in c_i which is satisfied by the interpreted truth valuation on the $k + 4$ th level. So, if the truth valuation satisfies at least one literal in clause c_i , we will have at least 4 of the branches connecting to g_i and h_i with which to extend to each of the 4 colors a_j . If our truth valuation satisfies none of the literals in some clause c_i , then the branches containing g_i and h_i will appear at most 3 times and we will not be able to realize one of the neighborhoods of the form $\langle g_i; h_i, a_j, r \rangle$. Note that we are also using the fact here that no literal appears with the same parity more than once in each clause since such a repetitive clause would only give us 3 or fewer such opportunities to extend to a_j .

Thus, we see that if G is an infinite tree realizing \mathcal{S} then the truth valuation given by level $k + 4$ as described earlier must be one which satisfies Φ .

Therefore, we have that Φ is satisfied if and only if \mathcal{S} is realized thus completing our reduction. \square

It is not difficult to extend this result for tree neighborhoods with larger radius, higher degree, or mixed degree.

3 Realizability via Finite Graphs

In this section, we examine the computational complexity of determining the existence of graphs (as opposed to trees without cycles) which witness the realization, consistency, or sufficiency of a set of tree neighborhoods. We showed in the previous section that the problem of determining the existence of an infinite tree from a set of tree neighborhoods of degree ≥ 3 is NP-complete. Every finite graph has, corresponding to it, an infinite tree formed by the infinite paths through the finite graph starting at a fixed root. Furthermore, unlike trees, the neighborhoods in graphs have higher dependencies on one another. Thus, we might expect the complexity of realizing a set of tree neighborhoods by a finite graph to be at least as “hard” as the problem of realizing a set of tree neighborhoods by an infinite tree. Surprisingly, we show in this chapter that this problem can be determined in polynomial time!

It is important to note that in order to preserve the “tree-like” qualities of our neighborhoods, we must put certain restrictions on the size of the cycles in the graph. If every ρ -neighborhood of a graph G belongs to \mathcal{S} , then G may not have any cycles of length less than $2\rho+2$.

Let \mathcal{S} be a set of tree neighborhoods of degree d and radius 1 with vertices colored by a set C of colors. We begin with determining the computational complexity of determining the satisfaction of \mathcal{S} via a finite graph and then prove results about the complexity of consistency and realization as corollaries.

3.1 Theorem. *The satisfaction of \mathcal{S} by a connected finite graph is decidable in polynomial time with respect to n (the number of neighborhoods in \mathcal{S}) and the number of colors from our set.*

One of the difficulties in satisfying a set of neighborhoods by a finite graph is finding the exact number of copies of each neighborhood from \mathcal{S} needed in order to “glue up” every edge while still guaranteeing that the neighborhood set of the graph is contained in \mathcal{S} . We will construct a matrix A corresponding to \mathcal{S} such that any nonnegative (non-trivial) integer solution to $AX = 0$ tells us how many copies of each neighborhood is needed to build a finite graph satisfying \mathcal{S} . If no such nonnegative integer solution X exists, we conclude that \mathcal{S} cannot be satisfied by a finite graph.

We begin by enumerating the colors and ordering them according to the enumeration. A *twig* t in a neighborhood N is an ordered triple (N, c, c') where c is the color of the center vertex of N and c' is the color of a leaf in N .

The matrix A is constructed as follows: The rows will correspond to unordered pairs of colors from our set. The columns will correspond to neighborhoods in \mathcal{S} . For $c \leq c'$

according to our ordering of the colors, the $a_{\{c,c'\},N}$ entry of the matrix will be given by:

$$a_{\{c,c'\},N} = \sum_{\text{twigs } (N,c,c')} 1 - \sum_{\text{twigs } (N,c',c)} 1.$$

Notice that if $c = c'$ then $a_{\{c,c'\},N} = 0$. Thus, we may eliminate the rows in A corresponding to pairs of identical colors. We will show that such symmetric twigs will never prevent us from building a finite graph realizing \mathcal{S} if such a graph exists.

We want to find a nonnegative non-trivial integer solution X to the equation $AX = 0$. Since $AX = 0$ is a system of homogeneous linear equations, we will either have only the trivial solution or infinitely many solutions. Thus, when referring to a particular solution, we will always specify the vector X unless it is obvious from the context. Note that any nonnegative non-trivial integer valued vector X prescribes a multiset of neighborhoods from \mathcal{S} . Let \mathcal{S}_X be the multi-set of the neighborhoods in this collection, \mathcal{T}_X be the multi-set of all twigs (both symmetric and non-symmetric) in \mathcal{S}_X , and let \mathcal{N}_X be the multi-set of the non-symmetric twigs in \mathcal{T}_X .

We define the relation M on a set of twigs such that $(N, c, c')M(N', d, d')$ if and only if the colors $c = d'$ and $c' = d$. In other words, M decides whether two neighborhoods may “glue up” along an edge with endpoint colors given by the twigs. Notice that M is a symmetric relation.

Let X be a non-trivial vector with only nonnegative integer entries. Then we may construct the graph \mathcal{M}_X as follows: Let the set of vertices be $V(\mathcal{M}_X) = \{t \in \mathcal{N}_X\}$ and the set of edges be $E(\mathcal{M}_X) = \{(t, t') : tMt'\}$.

Notice that \mathcal{M}_X is restricted to the non-symmetric twigs only and depends on which vector X we choose.

Recall that a *perfect matching* in a graph G is a set of edges from G such that every vertex of G is incident to exactly one edge. There has been much work done on the study of matchings. A result of particular interest to us is:

3.2 Theorem (Hall [5]). *Given a bipartite graph $G = (V_1, V_2, E)$, with the sets V_1 and V_2 of the same cardinality, there is a perfect matching in G if and only if for every subset $U \subseteq V_1$, the size of U is less than or equal to the size of the set of all vertices adjacent to at least one element of U .*

We will use Hall’s Theorem to prove the existence of a perfect matching in \mathcal{M}_X if and only if $AX = 0$. Then we will use the perfect matching in \mathcal{M}_X to construct a finite graph which will allow us to determine the satisfaction of \mathcal{S} .

3.3 Lemma. *Let \mathcal{S} be a set of n tree neighborhoods. Let X be a non-trivial $1 \times n$ vector with nonnegative integer entries. There is a perfect matching in \mathcal{M}_X if and only if $AX = 0$.*

Proof. Fix an arbitrary non-trivial $1 \times n$ vector X with nonnegative integer-valued entries.

We begin by showing that if $AX = 0$, then \mathcal{M}_X is a bipartite graph. For colors $c < c'$ according to the enumeration on C , let V_1 be the set of all non-symmetric twigs of the form (N_i, c, c') and let V_2 be the set of all non-symmetric twigs of the form (N_j, c', c) .

By the way that we set up the matrix A , the entry corresponding to $[AX]_{\{c,c'\}}$ is

$$\sum_{N \in \mathcal{S}} x_N a_{\{c,c'\},N} = \sum_{N \in \mathcal{S}} \sum_1^{x_N} a_{\{c,c'\},N} = \sum_{t \in \mathcal{N}_X} a_{\{c,c'\},N} = \sum_{t \in \mathcal{N}_X} \left(\sum_{t=(N,c,c')} 1 - \sum_{t=(N,c,c')} 1 \right).$$

So $[AX]_{\{c,c'\}} = 0$ if and only if the number of twigs of the form (N_i, c, c') in \mathcal{N}_X , is equal to the number of twigs of the form (N_j, c', c) in \mathcal{N}_X for $c \neq c'$. Thus, $AX = 0$ if and only if V_1 and V_2 have the same cardinality. If $(N_i, c, c') \in V_1$, then for all twigs t such that $(N_i, c, c')Mt$, t must have the form (N_j, c', c) and thus, it must be a member of V_2 . This implies that no edge has both its endpoints in the same set. Thus, \mathcal{M}_X is bipartite.

Since $AX = 0$ implies that the number of twigs of the form (N_i, c, c') in \mathcal{N}_X is equal to the number of twigs (N_j, c', c) in \mathcal{N}_X for each pair of distinct colors c, c' , we know that every vertex has degree at least the number of times its twig appears in \mathcal{N}_X . By Hall's Theorem, we know that there is a perfect matching in $(V_1, V_2, E(\mathcal{M}_X))$ if and only if for every subset $U \subseteq V_1$, the number of vertices in U is less than or equal to the number of vertices adjacent to a vertex from U . Since every vertex in V_1 is adjacent to at least as many vertices as its multiplicity in \mathcal{N}_X , we know that for any subset U , every vertex in U must be adjacent to at least as many vertices as its twig type appears in U . Therefore, Hall's condition is satisfied. Thus, if $AX = 0$, then \mathcal{M}_X has a perfect matching.

For the other direction, we see that if \mathcal{M}_X has a perfect matching, then for every twig (N_i, c, c') in \mathcal{N}_X , there is a twig (N_j, c', c) in \mathcal{N}_X . We showed that this is the case if and only if there is a vector X with nonnegative integer values such that $AX = 0$. \square

Next we will show that there is a perfect matching for \mathcal{M}_X if and only if there is a finite graph satisfying \mathcal{S} . We begin with some preliminary lemmas.

Given any finite graph G , we may construct a finite graph G' such that

- G' has no cycles of length less than or equal to $2\rho+1$,
- there is a vertex v adjacent to vertices colored by x_1, \dots, x_d in G if and only if there is a vertex colored by v and adjacent to vertices colored by x_1, \dots, x_d in G' , and
- G' is connected if and only if G is.

We construct G' as follows. Let m be the number of cycles of length less than or equal to $2\rho+1$ in G . Choose p to be the least prime number greater than m .

Enumerate the vertices of G . This gives an orientation to the edges of G ; $u \rightarrow v$ if $u < v$ in the enumeration. For each directed edge, we assign a shift value; $\sigma : E(G) \rightarrow \mathbb{Z}/p$ such that for all cycles of length $\leq 2\rho+1$, given by e_1, \dots, e_k , we have $\sum_{i=1}^k \sigma(e_i) \neq 0 \pmod{p}$. The following lemma shows that such a function σ exists.

3.4 Lemma. *There is a function $\sigma; E(G) \rightarrow \mathbb{Z}/p$ such that for all cycles of length $k \leq 2\rho+1$, given by e_1, \dots, e_k , it is the case that $\sum_{i=1}^k \sigma(e_i) \neq 0 \pmod{p}$.*

Proof. We construct the map σ one element of the domain at a time. With each assignment, we ensure that every cycle of length $\leq 2\rho+1$ containing that edge does not sum to $0 \pmod p$. Suppose there is an edge e such that there is no way to assign a value $\sigma(e)$ without a cycle of length $\leq 2\rho+1$ containing the edge e adding to $0 \pmod p$. This means that there are cycles of length $\leq 2\rho+1$ containing e which add up to $0, 1, 2, \dots$, and $p-1 \pmod p$ before we add in the value for $\sigma(e)$. We chose p such that there are fewer than p cycles of length $\leq 2\rho+1$. This gives us a contradiction. Thus, there is always a way to assign a shift value σ to each edge in G such that every cycle of length $\leq 2\rho+1$ in G , given by e_1, \dots, e_k , has the property that $\sum_{i=1}^k \sigma(e_i) \neq 0 \pmod p$. \square

Let the vertices of G' be the collection of all pairs (v, ℓ) where v is a vertex from G and $\ell \in \{0, 1, 2, \dots, p-1\}$. There will be an edge in G' from (u, ℓ) to $(v, \ell + \sigma \pmod p)$ if (u, v) is an edge in G , $u < v$ according to the enumeration on the vertices of G , $\ell \in \{0, 1, 2, \dots, p-1\}$, and the shift assigned to the edge (u, v) in G is σ . In other words, G' has p copies of the vertex set of G and edges G' are given by the shift function σ .

It is easy to check that G' has no cycles of length $\leq 2\rho+1$. Furthermore, the graph G' was constructed so that the adjacencies from G are preserved.

Now we show that G' is connected.

3.5 Lemma. *If the graph G is connected and has at least one cycle of length $\leq 2\rho - 1$, then the graph G' is connected.*

Proof. Since the sum of the values of the edges in any cycle of length $\leq 2\rho - 1$ must add to some nonzero number λ modulo p , a copy of the small cycle in G' will end λ many levels from where it began. Since p is a prime number, it has no proper divisors so the lifting of the small cycle will pass through every level set. Thus, assume that G is connected and has a small cycle. Choose arbitrary vertices (u, k) and (v, ℓ) in G' . We will show that there is a path in G' from (u, k) to (v, ℓ) .

When we project these vertices onto G , there is a path connecting u to v since G is connected. This path is possibly trivial. Let the sum of the shifts of this path be $\lambda \pmod p$. This means that if we can find a path from (u, k) to $(u, \ell - \lambda)$, then by following the lifting of the path in G , there is a path from $(u, \ell - \lambda)$ to (v, ℓ) .

Now, assuming that $k \neq \ell - \lambda$, we must find a path from (u, k) to $(u, \ell - \lambda)$ in G' . Let w be a vertex in the small cycle in G . Since G is connected, there is a path from u to w in G . Let the sum of the shifts of this path be $\kappa \pmod p$. This means that there is a path from (u, k) to $(w, k + \kappa)$ in G' . Since $(w, k + \kappa)$ is in the lifting of the small cycle, we know that there is a path from $(w, k + \kappa)$ to $(w, \ell - \lambda - \kappa)$ in G' . Thus, the path in G' from (u, k) to $(w, k + \kappa)$ to $(w, \ell - \lambda - \kappa)$ to $(u, \ell - \lambda)$ to (v, ℓ) is the desired path in G' . Therefore, G' is connected. \square

Notice that this proof required that G had at least one cycle of length $\leq 2\rho - 1$ in order to guarantee that G' is connected. This does not cause any problems since if G does not have a cycle of length $\leq 2\rho - 1$ then we have no need to construct the graph G' .

We define the *type* of a symmetric twig to be the color of its endpoints.

Note that if there is a solution to $AX = 0$ then there is a solution in which each type of symmetric twig appears an even number of times in \mathcal{T}_X . If X has the property that \mathcal{T}_X does not have an even number of each type of symmetric twig, then we consider the vector $2X$. If $AX = 0$, then $A(2X) = 0$ and clearly, every type of symmetric twig appears an even number of times in \mathcal{T}_{2X} . Since we showed in Lemma 3.3 that there is a nonnegative integer solution to $AX = 0$ if and only if there is a perfect matching in \mathcal{M}_X (which only relies on the non-symmetric twigs), we know that there is a perfect matching in \mathcal{M}_X if and only if there is a perfect matching in \mathcal{M}_{2X} .

3.6 Theorem. *For any non-trivial $1 \times n$ vector X with all nonnegative integer values, there is a perfect matching in \mathcal{M}_X if and only if there is a finite graph which satisfies \mathcal{S} .*

Proof. As in the discussion above, we know that if there is a non-trivial vector with nonnegative integer entries such that there is a perfect matching in the graph of the non-symmetric twigs, then there is a vector X such that every type of symmetric twig appears an even number of times in \mathcal{T}_X and there is a perfect matching in \mathcal{M}_X . Thus, we may assume that X is a vector with these properties. Since \mathcal{T}_X has an even number of each type of symmetric twig, we may match each symmetric twig with exactly one other symmetric twig of its type.

We construct the finite graph G as follows. Begin by enumerating the neighborhoods in \mathcal{S}_X . The vertices of G will be the set of all v_i such that v is the color of the center vertex of the i th neighborhood in the enumeration of \mathcal{S}_X . There is an edge from u_i to v_j if one of the two cases hold:

- The twig (N_i, u, v) is adjacent to the twig (N_j, v, u) in the perfect matching on \mathcal{M}_X ,
or
- The twigs (N_i, u, v) and (N_j, v, u) are symmetric twigs of the same type which we paired together.

The graph G is finite since the number of vertices in G is the same as the sum of the entries in X .

Suppose that there is a vertex in G colored by v and adjacent to vertices colored by x_1, \dots, x_d . Then v must be the center vertex of some neighborhood in \mathcal{S}_X (and thus, from \mathcal{S}).

Thus we have the existence of a finite graph G' which satisfies \mathcal{S} . Note that G and therefore G' may or may not be connected but that connectedness is irrelevant for satisfaction.

Now suppose that there is a graph G which satisfies \mathcal{S} . Let \mathcal{S}_X be the multi-set of neighborhoods in G and let X prescribe the number of each type of neighborhood in G . Enumerate the vertices of G . If u_i is adjacent to v_j in G , then we must have that $(N_i, u, v)M(N_j, v, u)$. Thus, G prescribes a perfect matching on \mathcal{M}_X . \square

Thus, we have shown:

3.7 Theorem. *If X is a nontrivial vector with nonnegative integer values such that the set \mathcal{T}_X has an even number of each type of symmetric twigs, the following are equivalent:*

1. X satisfies $AX = 0$,
2. There is a perfect matching on the set \mathcal{T}_X ,
3. There is a finite connected graph whose neighborhood set is contained in \mathcal{S} .

So there is a finite graph satisfying \mathcal{S} if and only if there is a non-trivial, nonnegative integer solution X to the equation $AX = 0$. An algorithm known as the Ellipsoid Method can be used to solve linear programming problems in polynomial time [6]. The solution set to a linear programming problem can be restricted to nonnegative integer solutions. Thus, the Ellipsoid Method allows us to check in polynomial time for the existence of a non-trivial, nonnegative integer valued vector satisfying $AX = 0$.

Thus, we have shown that the sufficiency of \mathcal{S} can be determined in polynomial time. Now we will look at the complexities of the same problem for realizability and consistency.

We have shown that there is a polynomial time algorithm for determining the sufficiency of \mathcal{S} by a finite graph. The proof showed that \mathcal{S} is sufficient if and only if there is a nonnegative (non-trivial) vector X such that $AX = 0$. The vector X prescribes the number of each type of neighborhood required to guarantee a perfect matching of all the twigs in \mathcal{T}_X . Thus, if we require that X has all positive integer entries, then there would be a perfect matching in which every neighborhood was represented at least once. The linear programming techniques used to determine the existence of such a vector X still run in polynomial time even if we require X to have only positive integer entries. We will use this idea to prove that realizability and consistency can be determined in polynomial time.

3.8 Theorem. *The consistency of \mathcal{S} by a finite graph can be determined in polynomial time with respect to n (the number of neighborhoods in \mathcal{S}) and the number of colors in C .*

Proof. We construct the matrix A just as before so that the $a_{\{c,c'\},N}$ entry will be given by:

$$a_{\{c,c'\},N} = \sum_{\text{twigs } (N,c,c')} 1 - \sum_{\text{twigs } (N,c',c)} 1.$$

Theorem 3.6 showed that there is a nontrivial nonnegative valued vector X satisfying $AX = 0$ if and only if there is a finite (possibly not connected) graph G witnessing every neighborhood prescribed by \mathcal{S}_X . We require X to have only positive integer entries. In this case, G must witness every neighborhood from \mathcal{S} at least once. Thus, for any neighborhood N in \mathcal{S} , the connected component containing N is a finite connected graph witnessing N and only exhibiting neighborhoods from \mathcal{S} . Conversely, if G is a graph which witnesses the neighborhood N and exhibits only neighborhoods from \mathcal{S} , then there is a non-trivial nonnegative integer valued vector X satisfying $AX = 0$. If we sum the vectors corresponding to every neighborhood N in \mathcal{S} , the resulting vector Y must satisfy $AY = 0$ and must have all positive integer entries. Thus, there is a vector X with positive integer entries satisfying $AX = 0$ if and only if \mathcal{S} is consistent. As before, we may determine, in polynomial time, the existence of such a vector X using linear programming techniques. \square

The algorithm for determining the realizability of \mathcal{S} will work in much the same way. Unfortunately, there is one problem with realizability: we require that the witnessing graph be connected (unlike sufficiency and consistency where we are only interested in realizing a connected component of the graph). The finite graph constructed from the perfect matching may or may not yield a connected graph. We can however, determine in polynomial time whether there is a connected graph. We will show this by determining in, polynomial time, the existence of a partial matching on \mathcal{M}_X and we will show that this partial matching can always be extended to a perfect matching on \mathcal{M}_X .

3.9 Lemma. *Let X be a vector with positive integer entries. Every partial matching in \mathcal{M}_X can be extended to a perfect matching in \mathcal{M}_X if X satisfies $AX = 0$.*

Proof. Suppose (V_1, V_2, E) is the bipartite graph description of \mathcal{M}_X from the proof of Lemma 3.3. Let $W_1 \subset V_1$ and $W_2 \subset V_2$ be subsets of vertices from \mathcal{M}_X such that there is a perfect matching on (W_1, W_2, E) ; in other words, let (W_1, W_2, E') describe a partial matching on \mathcal{M}_X . We will show that we may extend this partial matching to a perfect matching on \mathcal{M}_X by creating a perfect matching on $(V_1 - W_1, V_2 - W_2, E)$.

Let U be a subset of $V_1 - W_1$. Hall's condition requires that the size of the set of vertices adjacent to any vertex in U is greater than or equal to the size of U . Notice that since there is a perfect matching on (W_1, W_2, E') , we have that the size of W_1 is equal to the size of W_2 . Thus, the number of vertices in $V_1 - W_1$ is the same as the number of vertices in $V_2 - W_2$. Furthermore, if there are m vertices in $V_1 - W_1$ corresponding to twigs with endpoint colors c, c' , then there are m vertices in $V_2 - W_2$ corresponding to twigs with endpoints c', c . Thus, by Hall's Theorem, there is a perfect matching on $(V_1 - W_1, V_2 - W_2, E)$. Thus, we have extended the partial matching to a perfect matching on \mathcal{M}_X . \square

3.10 Theorem. *The realization of \mathcal{S} by a finite connected graph is decidable in polynomial time with respect to n (the number of neighborhoods in \mathcal{S}) and the number of colors in C .*

Proof. As before, we construct the matrix A corresponding to \mathcal{S} . Theorem 3.8 showed that there is a finite (possibly non-connected) graph G such that the set of neighborhoods in G is exactly \mathcal{S} if and only if there is a vector X with positive integer valued entries satisfying $AX = 0$. Furthermore, we know that determining the existence of such a vector X can be done in polynomial time.

We showed in section 2 that the de Bruijn graph \mathcal{B} can be constructed in polynomial time with respect to n and the size of C . We can thus check in polynomial time whether the de Bruijn graph is connected. If the de Bruijn graph is not connected then there can be no connected graph realizing \mathcal{S} since every path in \mathcal{B} corresponds to a path in a graph and vice versa. Thus, we assume that \mathcal{B} is connected. Since it is connected we can find a spanning tree in \mathcal{B} . Let the multi-set \mathcal{S}_Y be the collection of all neighborhoods prescribed by this spanning tree. From \mathcal{S}_Y we can build a connected piece of a finite graph which realizes \mathcal{S} . Suppose that X is a positive integer valued vector satisfying $AX = 0$. Let Z be a nonnegative valued vector such that $Y + Z$ is equal to some scalar multiple of X , say mX . We may assume that \mathcal{T}_{Y+Z} has an even number of each type of symmetric twig

since we may choose Z however we wish. Thus we have that $A(Y + Z) = 0$, and that there is a perfect matching on \mathcal{M}_{Y+Z} as well as on all the symmetric twigs.

Let the spanning tree of \mathcal{B} prescribe the partial matching on \mathcal{M}_{Y+Z} ; thus it gives a perfect matching on \mathcal{M}_Y . We showed in Lemma 3.9 that this partial matching may be extended to a perfect matching on \mathcal{M}_{Y+Z} . We then use this perfect matching to build a finite graph whose set of neighborhoods is exactly \mathcal{S} . This graph still may not be connected but we know that there is at least one component (the component containing every neighborhood from Y) which realizes \mathcal{S} . Thus, we may eliminate all other components and we have a connected graph G which realizes \mathcal{S} . Let W be the vector describing the multi-set of neighborhoods in the finite connected graph G . Note that G prescribes a perfect matching on \mathcal{M}_W . Thus, by Lemma 3.3 we have that W satisfies $AW = 0$. The vector W must have positive integer entries since every neighborhood appears at least once in the graph constructed from W .

Thus we have shown that the de Bruijn graph corresponding to \mathcal{S} is connected and there is a positive integer valued vector W such that $AW = 0$ if and only if there is a finite connected graph G which realizes \mathcal{S} . The connectedness of the de Bruijn graph and the existence of a vector W can be verified in polynomial time. Thus, we have that the realizability of \mathcal{S} can be verified in polynomial time. \square

4 Conclusion

Given a set \mathcal{S} of arbitrary neighborhoods (not necessarily tree neighborhoods), the problem of determining whether or not there is a graph realizing \mathcal{S} is recursively undecidable. We showed that by restricting the set \mathcal{S} to tree neighborhoods, we can improve the computational bounds. Furthermore, if we require the graph to be an infinite tree then the problem of determining the realizability of \mathcal{S} via such a graph is NP-complete. If however we wish to determine the realizability of \mathcal{S} via a finite graph, the problem is in P.

In the first author's exploration of the recursively undecidable general case, all the neighborhoods had many cycles and were quite dense. We continue to explore how the condition of allowing cycles in our neighborhoods affects the computational complexity of realizing or satisfying a set of neighborhoods. Some progress has been made in this realm but there is still more to be done.

References

- [1] Sanjeev Arora and Ronald Fagin. On winning strategies in Ehrenfeucht-Fraïssé games. *Theoret. Comput. Sci.*, 174(1-2):97–121, 1997.
- [2] V. K. Bulitko. Graphs with prescribed environments of the vertices. *Trudy Mat. Inst. Steklov.*, 133:78–94, 274, 1973. Mathematical logic, theory of algorithms and theory of sets (dedicated to P. S. Novikov on the occasion of his seventieth birthday).
- [3] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fund. Math.*, 49:129–141, 1960/1961.

- [4] Rachel M. Esselstein. *On the Complexity of Building a Graph with Given Neighborhoods*. Ph.D. Thesis. Dartmouth College, May 31, 2007.
- [5] P. Hall. On representatives of subsets. *Proc. London Math. Soc. (3)*, 10:26–30, 1935.
- [6] L. G. Khachian. A polynomial algorithm in linear programming. *Doklady Akademiia Nauk SSSR*, 244:1093–1096, 1979.
- [7] Thomas Schwentick. On winning Ehrenfeucht games and monadic NP. *Ann. Pure Appl. Logic*, 79(1):61–92, 1996.