Identifying Codes on Directed De Bruijn Graphs

Debra Boutin [*]
Department of Mathematics
Hamilton College

Victoria Horan [†]
Air Force Research Laboratory
Information Directorate

December 19, 2014

Abstract

For a directed graph G, a *t-identifying code* is a subset $S \subseteq V(G)$ with the property that for each vertex $v \in V(G)$ the set of vertices of S reachable from v by a directed path of length at most t is both non-empty and unique. A graph is called *t-identifiable* if there exists a t-identifying code. This paper shows that the de Bruijn graph $\vec{\mathcal{B}}(d,n)$ is 1- and 2-identifiable and examines conditions under which it is not t-identifiable. This paper also proves that a t-identifying code for t-identifiable de Bruijn graphs must contain at least $d^{n-1}(d-1)$ vertices. Constructions are given to show that this lower bound is achievable for 1-identifying codes when n is odd, or n is even and $d > 2$, and for 2-identifying codes when $n > 3$. Further a construction is given proving that when n is even and $d = 2$ there is a 1-identifying code of

[*] dboutin@hamilton.edu

[†] victoria.horan.1@us.af.mil

Approved for public release; distribution unlimited: 88ABW-2014-5829.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **19 DEC 2014**	2. REPORT TYPE	3. DATES COVERED **00-00-2014 to 00-00-2014**

4. TITLE AND SUBTITLE

Investigation of Zero Knowledge Proof Approaches Based On Graph Theory

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory,,Information Directorate,, , ,

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

For a directed graph G, a t-identifying code is a subset S ??? V (G) with the property that for each vertex v ??? V (G) the set of vertices of S reachable from v by a directed path of length at most t is both non-empty and unique. A graph is called t-identifiable if there exists a t-identifying code. This paper shows that the de Bruijn graph ~B (d, n) is 1- and 2-identifiable and examines conditions under which it is not t-identifiable. This paper also proves that a t-identifying code for t- identifiable de Bruijn graphs must contain at least dn???1(d???1) vertices. Constructions are given to show that this lower bound is achievable for 1-identifying codes when n is odd, or n is even and d > 2, and for 2-identifying codes when n > 3.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**21**	

size at most one more than the lower bound. Additionally this paper proves that for $\vec{\mathcal{B}}(d,n)$ the minimum size of a directed resolving set (a subset with the property that every vertex of the graph can be distinguished by its directed distances to vertices of S) is $d^{n-1}(d-1)$, and that if $d > n$ the minimum size of a *determining set* (a subset S with the property that the only automorphism that fixes S pointwise is the trivial automorphism) is $\lceil \frac{d-1}{n} \rceil$.

1 Introduction

First introduced in 1998 [8], an *identifying code* for a graph G is a subset $S \subseteq V(G)$ with the property that for each $v \in V(G)$ the subset of vertices of S that are adjacent to v is non-empty and unique. That is, each vertex of the graph is uniquely identifiable by the non-empty subset of vertices of S to which it is adjacent. Note that not all graphs have an identifying code; those that do are called *identifiable*. A graph fails to be identifiable if and only if it contains a pair of vertices with the same closed (in-)neighborhood; such vertices are called *twins*. Extending these definitions to t-identifying and t-twins is easy and is covered in Section 3. Identifying codes can be quite useful in applications. For example, an identifying code in a network of smoke detectors allows us to determine the exact location of a fire given only the set of detectors that have been triggered. However, the problem of finding identifying codes is NP-Hard [5]. The computational cost has so far limited the real-world use of identifying codes.

The directed de Bruijn graph $\vec{\mathcal{B}}(d,n)$ is a directed graph in which the vertices are strings of length n from an alphabet $\mathcal{A}$ with d letters, and with a directed arc from vertex $x_1x_2\dots x_n$ to vertex $x_2\dots x_na$ for each $a \in \mathcal{A}$. When looking for a graph model for applications, it is useful to choose a graph with relatively few edges, but many short paths between any pair of vertices [2]. The de Bruijn graphs have both of these desirable properties. In addition, given an arbitrary pair of vertices in a de Bruijn graph, there are routing algorithms that, with high probability, create a path of length $O(\log n)$ between the pair [11]. The properties of de Bruijn graphs enable some problems that are NP-complete on general graphs, such as the Hamilton cycle problem, to be computationally solvable on de Bruijn graphs [12]. We will see that for most de Bruijn graphs, the construction of minimum size 1- and 2-identifying codes is indeed solvable.

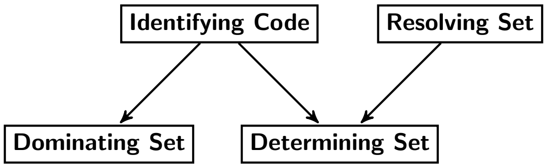

Figure 1: Vertex Subset Relationships

Other vertex subsets that we consider for de Bruijn graphs in this paper are dominating sets, determining sets, and resolving sets. Dominating sets provide complete coverage for a graph, while resolving sets provide an identification of vertices in graphs using relative distances. Finally, determining sets provide a set of vertices that is only fixed pointwise by the trivial automorphism. These types of subsets are also useful in applications. For example, resolving sets have been used in aiding the navigation of robots when distances to sufficient landmarks are known [9], and determining sets are useful in graph distinguishing which can reduce graph symmetry to enhance recognition. These different vertex subsets are interrelated, as shown in Figure 1. For example, each resolving set and identifying code is also a determining set, but not vice-versa. This is discussed more fully in Section 4.1 and Section 4.3.

In Section 2, we give careful definitions necessary for working with directed de Bruijn graphs. In Section 3, we prove that for every t-identifiable de Bruijn graphs, any t-identifying code must contain at least $d^{n-1}(d-1)$ vertices. We prove by construction that this bound is achievable for 1-identifying codes when n is odd, or n is even and $d > 2$, and for 2-identifying codes when $n > 3$. For even n with $d = 2$ we show the existence of a 1-identifying code of size $d^{n-1}(d-1)+1$, which is at most one more than optimal size. The existence of vertices with identical t-in-balls in the graph (called *t-twins*) guarantees that the graph has no t-identifying set. In Section 3.1 we address the question of when $\vec{\mathcal{B}}(d,n)$ contains t-twins. In Section 4 we study dominating sets, directed resolving sets, and determining sets for $\vec{\mathcal{B}}(d,n)$. Section 4.2 gives a proof that the minimum size of a directed resolving set for $\vec{\mathcal{B}}(d,n)$ is $d^{n-1}(d-1)$, and Section 4.3 that the minimum size of a determining set is $\lceil \frac{d-1}{n} \rceil$.

2 Definitions

We will be considering various types of vertex subsets on the class of directed de Bruijn graphs. The following definitions will be useful in working with this class of graphs.

Definition 2.1. Let $\boldsymbol{\mathcal{A}_d} = \{0, 1, \ldots, d-1\}$ and $\mathcal{A}_d^n$ set of all strings of length n made up of **letters** of $\mathcal{A}$. When d is clear from context we will use $\mathcal{A}$ and $\mathcal{A}^n$ respectively.

Definition 2.2. The **directed de Bruijn graph**, denoted $\boldsymbol{\vec{\mathcal{B}}(d, n)}$, has vertex set $\mathcal{A}_d^n$, with an edge from string $x_1x_2 \ldots x_n$ to string $y_1y_2 \ldots y_n$ exists if and only if $x_2x_3 \ldots x_n = y_1y_2 \ldots y_{n-1}$.

Definition 2.3. The **concatenation of two strings** $x = x_1x_2 \ldots x_i$ and $y = y_1y_2 \ldots y_k$ is given by $x \oplus y = x_1x_2 \ldots x_iy_1y_2 \ldots y_k$.

Definition 2.4. The **concatenation of sets of strings** S and T is given by $S \oplus T = \{x \oplus y \mid x \in S \text{ and } y \in T\}$.

Definition 2.5. The **prefix** of a string $x = x_1x_2 \ldots x_n$ is the substring $x_1x_2 \ldots x_{n-1}$, denoted by $\boldsymbol{x^-}$.

Definition 2.6. The **suffix** of a string $x = x_1x_2 \ldots x_n$ is is the substring $x_2x_3 \ldots x_n$, denoted by $\boldsymbol{x^+}$.

Definition 2.7. When discussing substrings of a string $x_1x_2 \ldots x_n$, we will use the notation $\boldsymbol{x(a : b)}$ to denote the substring $x_ax_{a+1} \ldots x_b$.

Definition 2.8. If a string $x_1x_2 \ldots x_n$ contains a constant substring, i.e. $x(1 : n) = 0$, then we will denote the consecutive letters as the constant raised to an exponent that denotes the length, i.e. $x = 0^n$. This will also be used for repeated substrings, such as $0101 \ldots 01 = (01)^{n/2}$.

3 Identifying Codes

We begin by building up to the definition of an identifying code. This requires careful definitions of directed distance and t-balls.

Definition 3.1. The **directed distance** from vertex u to vertex v in a graph G is given by $\vec{d}(u,v)$, and is defined as the length of the shortest directed path from u to v in G.

Definition 3.2. Let $v \in V(G)$. The **open in-neighborhood** of v is given by $\boldsymbol{N^-(v)} = \{u \in V(G) \mid (u,v) \in E(G)\}$, and the **closed in-neighborhood** is given by $\boldsymbol{N^-[v]} = N^-(v) \cup \{v\}$.

Note that out-neighborhoods are defined analogously, but will not be needed in this paper.

Definition 3.3. The **in-ball of radius** t centered at vertex v is the set: $\boldsymbol{B_t^-(v)} = \{u \in V(G) \mid \vec{d}(u,v) \leq t\}$. and the **out-ball of radius** t centered at vertex v is the set: $\boldsymbol{B_t^+(v)} = \{u \in V(G) \mid \vec{d}(v,u) \leq t\}$.

The two following lemmas are useful in working with distances in $\vec{\mathcal{B}}(d,n)$ and their proofs are self-evident.

Lemma 3.4. In $\vec{\mathcal{B}}(d,n)$ there is a directed path of length $t \leq n$ from x to y if and only if $x(t+1:n) = y(1:n-t)$. That is, if and only if the rightmost $n-t$ letters of x are the same as the leftmost $n-t$ letters of y.

Lemma 3.5. In $\vec{\mathcal{B}}(d,n)$ if vertices $x \neq y$ have the same prefix, then for all $u \neq \{x,y\}, \vec{d}(u,x) = \vec{d}(u,y)$. In particular, $B_t^-(x) \setminus \{x\} = B_t^-(y) \setminus \{y\}$ for all $t \leq n$.

Definition 3.6. Given a subset $S \subset V(G)$, the S t**-identifying set** for vertex v is given by $\mathsf{ID}_S(v) = B_t^-(v) \cap S$.

Definition 3.7. A t**-identifying code** is set $S \subseteq V(G)$ such that each vertex has a unique, non-empty identifying set. That is, for every $u \in V(G)$, $\mathsf{ID}_S(u) \neq \emptyset$, and for all pairs $u, v \in V(G)$ we have $\mathsf{ID}_S(u) \neq \mathsf{ID}_S(v)$. The variable t is referred to as the **radius** of the identifying code. See Figure 2 for an identifying code in the graph $\vec{\mathcal{B}}(2,3)$.

In the above definitions, if t is omitted from the notation (i.e. identifying code instead of t-identifying code), then it is assumed that $t = 1$.

Note that not every graph has a t-identifying code for each t. In particular if the graph has two vertices with equal in-balls of radius t, then the graph has no t-identifying code. The topic of such 't-twins' and the resulting non-existence of t-identifying codes is covered in Section 3.1.

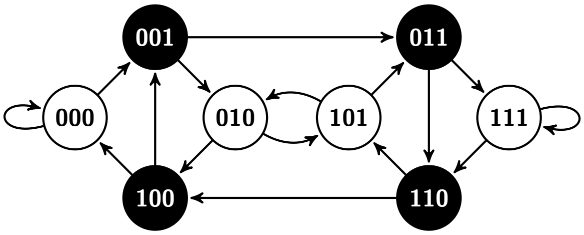

Figure 2: A 1-identifying code in the graph $\vec{\mathcal{B}}(2,3)$ (black vertices). A 2-identifying code in this graph requires all vertices but one of $\{000, 111\}$, and there are no t-identifying codes for $t \geq 3$.

Theorem 3.8. *If $\vec{\mathcal{B}}(d,n)$ is a t-identifiable graph, then the size of any t-identifying code is at least $d^{n-1}(d-1)$.*

Proof. Choose $t \leq n$ and $a \neq b$ in $\mathcal{A}$. Suppose that for some $w \in \mathcal{A}^{n-1}$, niether $x = w \oplus a$ nor $y = w \oplus b$ is a set S. Since x and y share a prefix, by Lemma 3.5, $B_t^-(x) \setminus \{x\} = B_t^-(y) \setminus \{y\}$. Since neither x nor y is in S, $\mathsf{ID}_S(x) = \mathsf{ID}_S(y)$. Thus S is not a t-identifying code. Thus for each $w \in \mathcal{A}^{n-1}$, a t-identifying code must contain, at least, all but one of $w \oplus a$ for $a \in \mathcal{A}$. Thus a t-identifying code for $\vec{\mathcal{B}}(d,n)$ must have size at least $d^{n-1}(d-1)$. □

Note that the result above is independent of the radius t. An interesting consequence of this is the fact that increasing the radius of our identifying code does not produce any decrease in the size of a minimum identifying code. For example, consider the potential application of identifying codes in sensor networks. One might think that by increasing the sensing power (which corresponds to the radius of the identifying code) we would be able to place fewer sensors and thus incur a savings overall. However, Theorem 3.8 implies that providing more powerful (and thus more expensive) sensors does not allow us to place fewer sensors. Thus we should use sensors that have sensing distance equivalent to radius one. In fact, in the case of 2-identifying codes in $\vec{\mathcal{B}}(2,3)$, we actually require an extra vertex for a minimum size of seven! A more general result for 2-identifying codes for $n > 3$ is given in Theorem 3.11. First, we will look at examples of optimal and almost-optimal 1-identifying codes.

Theorem 3.9. *If n is odd, or n is even and $d > 2$, then*

$$S = \mathcal{A}_d^n \setminus \{a \oplus \mathcal{A}_d^{n-2} \oplus a \mid a \in \mathcal{A}_d\}$$

is an identifying code for $\vec{\mathcal{B}}(d,n)$. Further this identifying code has optimal size $(d-1)d^{n-1}$.

Proof. Define S as in the statement of the theorem. First, we will see that the identifying set for every vertex has size either d or $d-1$. Let $x = x_1x_2\ldots x_n$, then

$$N^-(x) \cap S = \{\mathcal{A} \oplus x_1x_2\ldots x_{n-1}\} \setminus \{x_{n-1}x_1x_2\ldots x_{n-1}\}.$$

If $x_1 = x_n$, then $\mathsf{ID}_S(x) = N^-(x) \cap S$ has size $d-1$. Whereas, if $x_1 \neq x_n$, then $\mathsf{ID}_S(x) = \{x\} \cup N^-(x) \cap S$ has size d.

From this it is clear that every vertex has a non-empty identifying set. However we must also show that every identifying set is unique. Suppose there are two distinct vertices $x, y \in V(\vec{\mathcal{B}}(d,n))$ such that $\mathsf{ID}_S(x) = \mathsf{ID}_S(y)$. Call their identical identifying set T. We look at the two cases, $|T| = d$ and $|T| = d-1$, separately below.

Suppose that $|T| = d$. Then $\{x, y\} \subseteq T$ by our assumption on T and our earlier reasoning. Since $x \neq y$, this means that $\vec{\mathcal{B}}(d,n)$ contains both directed arcs $x \to y$ and $y \to x$. This allows us to conclude that $\{x,y\} = \{(ab)^k, (ba)^k\}$ for some distinct $a, b \in \mathcal{A}$ with $k = n/2$. In particular we must have n even. Below are the precise identifying sets for x and y.

$$\begin{aligned} \mathsf{ID}_S((ab)^k) &= \{(ab)^k, (ba)^k\} \cup \{c(ab)^{k-1}a \mid c \in \mathcal{A} \setminus \{a,b\}\} \\ \mathsf{ID}_S((ba)^k) &= \{(ab)^k, (ba)^k\} \cup \{c(ba)^{k-1}b \mid c \in \mathcal{A} \setminus \{a,b\}\} \end{aligned}$$

If $d > 2$ these two identifying sets are in fact different, which is a contradiction.

Suppose that $|T| = d-1$. Then neither x nor y is in T, which means neither is in S. However since their identifying codes are identical, this means that they have identical first neighborhoods. By definition of first neighborhoods, this means that x and y have the same prefix but different final letters. By then definition of S, one of x, y (if not both) is a member of S, which is a contradiction. □

Theorem 3.10. *If n is even, then*

$$S' = \{00(10)^{k-1}\} \cup \mathcal{A}_2^n \setminus \{a \oplus \mathcal{A}_2^{n-2} \oplus a \mid a \in \mathcal{A}_2\}$$

is an identifying code for $\vec{\mathcal{B}}(2,n)$. Further, this identifying code is within one of being optimal.

Proof. We showed in the proof of Theorem 3.9 that if $x \neq y$ have the property that $B_1^-(x) \cap S = B_1^-(y) \cap S$ with n even and $d = 2$ then we must have $\{x, y\} = \{(ab)^k, (ba)^k\}$ for $a \neq b \in \mathcal{A}_2$. By adding $00(10)^{k-1}$ to achieve S', we are either adding an in-neighbor of x or of y (but not both) that was not in S. This yields identifying sets of different size for x and for y. Note that either vertex from $00(10)^{k-1}, 11(01)^{k-1}$ would work to supplement S in this case. □

Note that while Theorem 3.10 provides an *almost* optimal solution, 1-identifying codes of size 2^{n-1} are indeed possible in $\vec{\mathcal{B}}(2, n)$ with n even. For example, the following set is a 1-identifying code of size 8 in $\vec{\mathcal{B}}(2, 4)$:

$$\{0001, 0010, 0100, 0111, 1000, 1011, 1101, 1110\}.$$

The following constructs 2-identifying codes for all $\vec{\mathcal{B}}(d, n)$ with $n > 3$.

Theorem 3.11. *Let $n > 3$ and $S = \mathcal{A}_d^n \setminus \{x_1 a x_3 x_4 \ldots x_{n-1} a \mid a \in \mathcal{A}_d\}$. If n is even, then S is a 2-identifying code for $\vec{\mathcal{B}}(d, n)$. If n is odd, then $S' = (S \cup \{(ab)^{\frac{n-1}{2}} b \mid \; a \neq b \in \mathcal{A}_2\}) \setminus \{(ab)^{\frac{n-1}{2}} a \mid \; a \neq b \in \mathcal{A}_2\}$ is a 2-identifying code for $\vec{\mathcal{B}}(d, n)$. In both these cases, the 2-identifying code is of optimal size $d^{n-1}(d-1)$.*

Proof. Consider an arbitrary string $x = x_1 x_2 x_3 \ldots x_n \in \mathcal{A}_d^n$, and define the set $T = \mathsf{ID}_S(x)$. We'll consider the contents of T in four cases based on the equality of x_1, x_{n-1} and of x_2, x_n. First let $C = \{a x^{--} \mid a \in \mathcal{A} \setminus \{x_{n-2}\}\}$.

Case 1. If $x_2 = x_n$ and $x_1 = x_{n-1}$, then $T = \mathcal{A} \oplus C$. Thus $|T| = d^2 - d$.

Case 2. If $x_2 \neq x_n$ and $x_1 = x_{n-1}$, then $T = (\mathcal{A} \oplus C) \cup \{x\}$. If $x \in \mathcal{A} \oplus C$ then $x^+ = a x^{--}$ for some $a \in \mathcal{A} \setminus \{x_{n-2}\}$. In this case,we have $x_2 x_3 \cdots x_n = a x_1 x_2 \cdots x_{n-2}$. This implies that we have $x_1 = x_3 = x_5 = \cdots$, and also that $x_2 = x_4 = x_6 = \cdots$. Since this case requires that $x_1 = x_{n-1}$, we must have that either n is even or that $x_1 = x_2 = x_3 = \cdots = x_n$. In either case, this contradicts our assumption that $x_2 \neq x_n$. Thus $x \notin \mathcal{A} \oplus C$, and we conclude that $|T| = d^2 - d + 1$.

Case 3. If $x_2 = x_n$ and $x_1 \neq x_{n-1}$, then $T = \mathcal{A} \oplus \{C \cup \{x^-\}\}$. If $x^- = a x^{--}$ for some $a \in \mathcal{A} \backslash \{x_{n-2}\}$, then $a x_1 x_2 \cdots x_{n-2} = x_1 x_2 \cdots x_{n-1}$. This implies that we have $x_1 = x_2 = x_3 = \cdots = x_{n-2} = x_{n-1}$. This contradicts our assumption that $x_1 \neq x_{n-1}$. Thus $x^- \neq a x^{--}$ for any $a \in \mathcal{A} \setminus \{x_{n-2}\}$, and we conclude that $|T| = d^2$.

Case 4. If $x_2 \neq x_n$ and $x_1 \neq x_{n-1}$, then $T = (\mathcal{A} \oplus \{C \cup \{x^-\}\}) \cup \{x\}$. As in Case 3, since $x_1 \neq x_{n-1}, \mathcal{A} \oplus \{C \cup \{x^-\}\}$ contains d^2 distinct elements. Let us consider whether $x \in \mathcal{A} \oplus \{C \cup \{x^-\}\}$. If not, then $|T| = d^2 + 1$. There are two cases to consider.

a. If $x \in \mathcal{A} \oplus C$, then $x^+ = x^-$. In this case, we must have that $x_2 x_3 x_4 \cdots x_n = x_1 x_2 \cdots x_{n-1}$, which implies that we have the following chain of equalities: $x_1 = x_2 = x_3 = \cdots = x_{n-1} = x_n$. This contradicts the assumptions that $x_2 \neq x_n$ and $x_1 \neq x_{n-1}$. Thus, this case does not occur.

b. If $x \in \mathcal{A} \oplus \{x^-\}$, then $x^+ = ax^{--}$ for some $a \in \mathcal{A}$. Then $x_2 x_3 \cdots x_n = a x_1 x_2 \cdots x_{n-2}$. This implies that $x_1 = x_3 = x_5 = \cdots$, and also that $x_2 = x_4 = x_6 = \cdots$. If n is even, this contradicts our assumptions that $x_2 \neq x_n$ and $x_1 \neq x_{n-1}$. Thus for even n, this case does not occur. For n odd, this case only occurs if $x \in \{(ab)^{\frac{n-1}{2}}a\}$.

Thus, if n is even, or n is odd and $x \notin \{(ab)^{\frac{n-1}{2}}a\}$, we can see that $T = \mathsf{ID}_S(x)$ completely determines the string x. In particular, given T we can decide which case we are in based on $|T|$. We can then determine $x_1, \ldots, x_n$ based on the content of T. Thus in these cases S is an identifying code.

However, if n is odd, and $x \in \{(ab)^{\frac{n-1}{2}}a\}$ we must change S to get an identifying code. Note that $B_2^-((ab)^{\frac{n-1}{2}}a) \cup \{(ab)^{\frac{n-1}{2}}b\} = B_2^-((ab)^{\frac{n-1}{2}}b)$. Since our set S contains vertices of the form $(ab)^{\frac{n-1}{2}}a$ but not $(ab)^{\frac{n-1}{2}}b$, these two types of vertices must have identical identifying sets with respect to S. Thus by adding the vertices in $\{(ab)^{\frac{n-1}{2}}b\}$, we are able to create distinct identifying sets with respect to $S \cup \{(ab)^{\frac{n-1}{2}}b\}$. However, we note that we now have the vertices of $\{(ab)^{\frac{n-1}{2}}b\}$ and $\{(ab)^{\frac{n-1}{2}}a\}$ in our identifying code, but that $B_2^+((ab)^{\frac{n-1}{2}}a) \cup \{b(ba)^{\frac{n-1}{2}}\} = B_2^+(b(ba)^{\frac{n-1}{2}})$. This implies that the inclusion of both $(ab)^{\frac{n-1}{2}}b$ and $(ab)^{\frac{n-1}{2}}a$ in our identifying code is only necessary if they are required to identify vertex $(ab)^{\frac{n-1}{2}}a$ from vertex $b(ba)^{\frac{n-1}{2}}$. So, as long as we can identify $(ab)^{\frac{n-1}{2}}a$ differently from $b(ba)^{\frac{n-1}{2}}$ without using $b(ba)^{\frac{n-1}{2}}$, we need only include $(ab)^{\frac{n-1}{2}}b$ and not $(ab)^{\frac{n-1}{2}}a$ in our identifying code. Since these two vertices have disjoint in-balls of radius 2 for $n > 3$, they must have distinct 2-identifying sets. Thus S' is a 2-identifying code in this case. □

3.1 Non-Existence Results

While we have just seen 1- and 2-identifying codes for many de Bruijn graphs, not all de Bruijn graphs have t-identifying codes for all t. For example, there are no 4-identifying codes for $\vec{\mathcal{B}}(2,5), \vec{\mathcal{B}}(3,5)$, and $\vec{\mathcal{B}}(3,6)$, among others. The problem of existence and non-existence of t-identifying codes in $\vec{\mathcal{B}}(d,n)$ is explored in this subsection.

Definition 3.12. Two vertices $u, v \in V(G)$ are called ***t*-twins** whenever $B_t^-(u) = B_t^-(v)$. If the graph has no t-twins, then G is called ***t*-twin-free**.

Theorem 3.13. [4] *For a given graph G and integer t, G has a t-identifying code if and only if it is t-twin-free.*

From this theorem, we can immediately determine an upper bound on t for $\vec{\mathcal{B}}(d,n)$ to be t-identifiable.

Corollary 3.14. *The graph $\vec{\mathcal{B}}(d,n)$ does not have a t-identifying code for any $t \geq n$.*

Proof. For $t \geq n$, we note that $B_t^-(x) = V(\vec{\mathcal{B}}(d,n))$, and so all vertices in our graph are twins. □

The following theorem illustrates another difficulty in attempting to use identifying codes of larger radius.

Theorem 3.15. *When n is even, there is no $(n-1)$-identifying code for $\vec{\mathcal{B}}(d,n)$.*

Proof. Consider vertices $x = (01)^{\frac{n-2}{2}}01$ and $y = (01)^{\frac{n-2}{2}}00$ in $\vec{\mathcal{B}}(d,n)$. Both of $B_{n-1}^-(x)$ and $B_{n-1}^-(y)$ consist of all strings that end in either 0 or 01. Thus these vertices are $(n-1)$-twins and $\vec{\mathcal{B}}(d,n)$ has no $(n-1)$-identifying code. □

Note that this only works for n even. Suppose n is odd and $x = (01)^{\frac{n-1}{2}}0$ and $y = (01)^{\frac{n-1}{2}}1$. Since $y \in B_{n-1}^-(y)$ but $x \notin B_{n-1}^-(y)$, these vertices can be distinguished by including y in S.

Theorem 3.16. *In $\vec{\mathcal{B}}(d,n)$, if x and y are t-twins, they are also $(t+1)$-twins.*

Proof. Suppose that $z \in B_{t+1}^-(x)$. Then we know that either $z \in B_t^-(x)$ or $z \in B_{t+1}^-(x) \setminus B_t^-(x)$. If $z \in B_t^-(x)$, then since x and y are t-twins, $z \in B_t^-(y) \implies z \in B_{t+1}^-(y)$. If $z \in B_{t+1}^-(x) \setminus B_t^-(x)$, $\vec{d}(z,x) = t+1$ so there exists $w \in B_t^-(x)$ with an arc from z to w. But again, since x and y are t-twins, $w \in B_t^-(y) \implies z \in B_{t+1}^-(y)$. Thus $B_{t+1}^-(x) \subseteq B_{t+1}^-(y)$. The same argument shows that $B_{t+1}^-(y) \subseteq B_{t+1}^-(x)$. Hence we have equality, so x and y are $(t+1)$-twins in $\vec{\mathcal{B}}(d,n)$. □

Corollary 3.17. *If there is no t-identifying code in $\vec{\mathcal{B}}(d,n)$, then there is no $(t+1)$-identifying code in $\vec{\mathcal{B}}(d,n)$.*

Theorem 3.18. *Let x and y be t-twins in $\vec{\mathcal{B}}(d,n)$. Define k to be the least integer such that (1) $k \geq 3$, and (2) $\{x,y\} \subseteq \mathcal{A}_k^n$. Then for all $\delta \geq k$, x and y are t-twins in $\vec{\mathcal{B}}(\delta,n)$.*

Proof. We have two cases. We define the notation $B_t^-(x,d) = B_t^-(x) \cap \mathcal{A}_d^n$.

Case 1: $k \leq \delta \leq d$. Note that $\mathcal{A}_\delta^n \subseteq \mathcal{A}_d^n$, so we have the following.

$$\begin{aligned}
B_t^-(x,\delta) &= B_t^-(x) \cap \mathcal{A}_\delta^n \\
&= B_t^-(x) \cap (\mathcal{A}_\delta^n \cap \mathcal{A}_d^n) \\
&= \left(B_t^-(x) \cap \mathcal{A}_d^n\right) \cap \mathcal{A}_\delta^n \\
&= B_t^-(x,d) \cap \mathcal{A}_\delta^n \\
&= B_t^-(y,d) \cap \mathcal{A}_\delta^n \\
&= B_t^-(y,\delta)
\end{aligned}$$

Thus x and y are t-twins in $\vec{\mathcal{B}}(\delta,n)$.

Case 2: $k \leq d \leq \delta$. Let $z \in \mathcal{A}_\delta^n$, and define $\alpha = \vec{d}(z,x)$ (distance from z to x) and $\beta = \vec{d}(z,y)$. Without loss of generality, assume that $\alpha \geq \beta$. Then we must have that $z_{\beta+1}z_{\beta+2}\ldots z_n \in \mathcal{A}_k^{n-\beta}$.

Select $w \in \mathcal{A}_k^n$ such that $w_{\beta+1}w_{\beta+2}\ldots w_n = z_{\beta+1}z_{\beta+2}\ldots z_n$, and so that the following hold for the first β letters of w: for each $i \in [\beta]$, we have $w_i \in \mathcal{A}_k \setminus \{x_1, y_1\}$. Note that this is possible since $k \geq 3$. By the design of w, we have ensured that $\vec{d}(w,x) = \vec{d}(z,x) = \alpha$ and $\vec{d}(w,y) = \vec{d}(z,y) = \beta$ in the graph $\vec{\mathcal{B}}(k,n)$. Since x and y are t-twins in $\vec{\mathcal{B}}(k,n)$, we know that either both $\alpha, \beta > t$, or both $\alpha, \beta \leq t$.

Returning to our original definition of α and β, this implies that either z is in both $B_t^-(x,\delta)$ and $B_t^-(x,\delta)$, or z is in neither t-in-ball. Since z was arbitrary, this implies that x and y are t-twins in $\vec{\mathcal{B}}(\delta,n)$. □

Corollary 3.19. *If there is no t-identifying code in $\vec{\mathcal{B}}(d,n)$ and $d \geq 3$, then there is no t-identifying code in $\vec{\mathcal{B}}(\delta,n)$ for any $\delta \geq d$.*

We now will prove a converse theorem.

Theorem 3.20. *Let x and y be t-twins in $\vec{\mathcal{B}}(d,n)$, and let $\varphi : \mathcal{A}_d \to \mathcal{A}_{d-1}$ be a surjection. If $\varphi(x) \neq \varphi(y)$, then $\varphi(x)$ and $\varphi(y)$ are t-twins in $\vec{\mathcal{B}}(d,n)$.*

Proof. We will first prove that $\varphi(B_t^-(x,d)) = B_t^-(\varphi(x), d-1)$. Once this is done, then we will have the following.

$$\begin{aligned} B_t^-(\varphi(x), d-1) &= \varphi(B_t^-(x,d)) \\ &= \varphi(B_t^-(y,d)) \\ &= B_t^-(\varphi(y), d-1). \end{aligned}$$

Hence, by proving the key equality, and by assuming that $\varphi(x) \neq \varphi(y)$, we will have produced a pair of t-twins in $\vec{\mathcal{B}}(d-1,n)$. We now prove the equality by showing containment in both directions.

Let $z \in \varphi(B_t^-(x,d))$. Then there exists some $\zeta \in \vec{\mathcal{B}}(d,n)$ such that $\zeta \in B_t^-(x,d)$ and $\varphi(\zeta) = z$. Hence $\vec{d}(\zeta, x) = a \leq t$ for some nonnegative integer a, and so $\zeta(a+1 : n) = x(1 : n-a)$. Since these substrings are equal, we also must have $\varphi(\zeta(a+1:n)) = \varphi(x(1:n-a)) = \varphi(x)(1:n-a)$. Thus $\vec{d}(\varphi(\zeta), \varphi(x)) \leq a \leq t$ in $\vec{\mathcal{B}}(d-1,n)$, and so $z \in B_t^-(\varphi(x), d-1)$, and therefore $\varphi(B_t^-(x,d)) \subseteq B_t^-(\varphi(x), d-1)$.

For the reverse, let $z \in B_t^-(\varphi(x), d-1)$, and then we suppose that $\vec{d}(z, \varphi(x)) = b \leq t$ for some nonnegative integer b. This then tells us that $z(b+1 : n) = \varphi(x)(1 : n-b) = \varphi(x(1 : n-b))$, which are over the alphabet $\{0, 1, 2, \ldots, d-2\}$. We construct a string $\zeta \in \vec{\mathcal{B}}(d,n)$ such that $\zeta(1 : b) = \varphi(z(1 : b))$ and $\zeta(b+1 : n) = x(1 : n-b)$. From this construction, it is clear that $\vec{d}(\zeta, x) \leq b \leq t$ and $\varphi(\zeta) = z$. Hence $\zeta \in B_t^-(x,d)$ and thus $z \in \varphi(B_t^-(x,d))$. Therefore $B_t^-(\varphi(x), d-1) \subseteq \varphi(B_t^-(x,d))$, and so the equality is shown. □

Corollary 3.21. *If there is no t-identifying code in $\vec{\mathcal{B}}(d,n)$ and $d \geq 3$, then there is no t-identifying code for $\vec{\mathcal{B}}(d-1,n)$.*

Proof. If there is no t-identifying code in $\vec{\mathcal{B}}(d,n)$, then there must exist a pair of t-twins in the graph - call them x and y. Using Lemma 3.20, to find a pair of t-twins in $\vec{\mathcal{B}}(d-1,n)$, we need only show that there exists a surjection $\varphi : \mathcal{A}_d \to \mathcal{A}_{d-1}$ such that $\varphi(x) \neq \varphi(y)$. Since $x \neq y$, there must be some index i such that $x_i \neq y_i$. Since $d-1 \geq 2$, we can choose a surjection φ such that $\varphi(x_i) \neq \varphi(y_i)$, and thus $\varphi(x) \neq \varphi(y)$. Then $\varphi(x)$ and $\varphi(y)$ are t-twins in $\vec{\mathcal{B}}(d-1,n)$, and so there is no t-identifying code. □

Combining Corollaries 3.19 and 3.21, we arrive at the following theorem.

Theorem 3.22. *Let $n, d \in \mathbb{Z}^+$ such that $d \geq 3$. There exists a t-identifying code in $\vec{\mathcal{B}}(d,n)$ if and only if there exists a t-identifying code in $\vec{\mathcal{B}}(d+1,n)$. Moreover, if a t-identifying code exists for $\vec{\mathcal{B}}(2,n)$, then a t-identifying code exists for $\vec{\mathcal{B}}(d,n)$.*

4 Dominating, Resolving, and Determining Sets

In this section we examine other types of vertex sets which identify or classify vertices up to some graph property. The properties used to define these sets are adjacency, distance, and automorphisms.

4.1 Dominating Sets

Definition 4.1. A (directed) ***t*-dominating set** is a subset $S \subseteq V(G)$ such that for all $v \in V(G)$ we have $B_t^-(v) \cap S \neq \emptyset$. That is, S is a (directed) t-dominating set if every vertex in G is within (directed) distance t of some vertex in S.

Note that by definition every identifying code is also a dominating set, but not conversely.

Theorem 4.2. [10] *For $d \geq 2$, $n \geq 1$, $dom(\vec{\mathcal{B}}(d,n)) = \lceil \frac{d^n}{d+1} \rceil$.*

In [10] a construction of a minimum dominating set for $\vec{\mathcal{B}}(d,n)$ is given. Key to this construction is the integer x defined by:

$$x = \begin{cases} d^{n-2} + d^{n-4} + \cdots + d^{n-2k} + \cdots + d^2 + 1 \bmod d^n, & \text{if } n \text{ is even;} \\ d^{n-2} + d^{n-4} + \cdots + d^{n-2k} + \cdots + d^3 + d \bmod d^n, & \text{if } n \text{ is odd.} \end{cases}$$

Let $D = \{x, x+1, \ldots, x + \lceil \frac{d^n}{d+1} \rceil - 1\}$. Now let S be the set of strings in $\mathbb{Z}_d^n$ that correspond (base d) to the integers in D. Then S is a minimum size dominating set for $\vec{\mathcal{B}}(d,n)$.

We also provide the following lower bound for the size of a minimum t-dominating set, and initial testing shows that this is a good estimate on size.

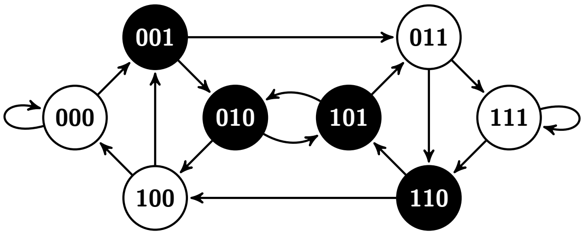

Figure 3: A directed resolving set in the graph $\vec{\mathcal{B}}(2,3)$ (black vertices).

Theorem 4.3. *Fix $d, n, t \in \mathbb{Z}^+$. Then we have the following lower bound on the size of a t-dominating set.*

$$dom^t(\vec{\mathcal{B}}(d,n)) \geq \left\lceil \frac{d^n}{d^t+1} \right\rceil$$

Proof. The graph $\vec{\mathcal{B}}(d,n)$ has d^n vertices, and the maximum size of a t-outball is $d^t + 1$. This gives us a lower bound as shown. □

4.2 Resolving Sets

Definition 4.4. A **directed resolving set** is a set S such that for all $u, v \in V(G)$ there exist $s \in S$ so that $\vec{d}(s,u) \neq \vec{d}(s,v)$. The **directed metric dimension** is the minimum size of a directed resolving set. An example of a directed resolving set in $\vec{\mathcal{B}}(2,3)$ is given in Figure 3.

Note that this definition is not quite the same as that given in [6] (which requires that there exist $s \in S$ so that $\vec{d}(u,s) \neq \vec{d}(v,s)$). Our definition corresponds better to the definitions of domination and of identifying codes for directed graphs that are used in this paper.

Theorem 4.5. *The directed metric dimension for $\vec{\mathcal{B}}(d,n)$ is $d^{n-1}(d-1)$.*

Proof. The following shows that for each $w \in \mathcal{A}^{n-1}$ a directed resolving set for $\vec{\mathcal{B}}(d,n)$ must contain (at least) all but one of the vertices with prefix w. Suppose that $w \in \mathcal{A}^{n-1}$, and $i \neq j \in \mathcal{A}$ so that neither of $w \oplus i, w \oplus j$ is in our set S. Note that if $x, y \in V(\vec{\mathcal{B}}(d,n))$, with $x \neq y$, then the distance from x to y is completely determined by x^- (and y^+). Since neither $w \oplus i$ nor $w \oplus j$ is in S, and both have the same prefix, $\vec{d}(w \oplus i, x) = \vec{d}(w \oplus j, x)$ for all $x \in S$. Thus S is not a directed resolving set. Thus for every $w \in \mathcal{A}^{n-1}$, S must contain (at least) all but one of the strings $w \oplus j$ for $j \in \mathcal{A}$. Thus $|S| \geq d^{n-1}(d-1)$. Since $\{w \oplus 0 \mid w \in \mathcal{A}^{n-1}\}$ can easily be shown to be a directed resolving set, we have the desired equality. □

The combination of Theorem 3.8 and Theorem 4.2 yields:

Corollary 4.6. *The directed metric dimension for $\vec{\mathcal{B}}(d,n)$ is equal to the minimum size of a t-identifying code for $\vec{\mathcal{B}}(d,n)$.*

4.3 Determining Sets

In this section we will use a determining set to help us illustrate the automorphism group of $\vec{\mathcal{B}}(d,2)$, study the relationship between $\mathrm{Aut}(\vec{\mathcal{B}}(d,n-1))$ and $\mathrm{Aut}(\vec{\mathcal{B}}(d,n))$ and use the result to find the determining number for each $\vec{\mathcal{B}}(d,n)$. First let's recall some definitions.

Definition 4.7. An **automorphism** of a graph G is a permutation π of the vertex set such that for all pairs of vertices $u, v \in V(G)$, uv is an edge between u and v if and only if $\pi(u)\pi(v)$ is an edge between $\pi(u)$ and $\pi(v)$. An **automorphism** of a directed graph G is a permutation π of the vertex set such that for all pairs of vertices $u, v \in V(G)$, uv is an edge from u to v if and only if $\pi(u)\pi(v)$ is an edge from $\pi(u)$ to $\pi(v)$. One automorphism in the binary (directed or undirected) de Bruijn graph is a map that sends each string to its complement.

Definition 4.8. [3] A **determining set** for G is a set S of vertices of G with the property that the only automorphism that fixes S pointwise is the trivial automorphism. The **determining number** of G, denoted $\mathrm{Det}(G)$ is the minimum size of a determining set for G. See Figure 4 for an example.

Note that an alternate definition for a determining set is a set S with the property that whenever $f, g \in \mathrm{Aut}(G)$ so that $f(s) = g(s)$ for all $s \in S$, then $f(v) = g(v)$ for all $v \in V(G)$. That is, every automorphism is completely determined by its action on a determining set.

Notice that since for both directed resolving sets and for identifying codes, since each vertex in a graph is *uniquely* identified by its relationship to the subset by properties preserved by automorphisms, the subset it also a determining set. Thus every directed resolving set and every identifying code

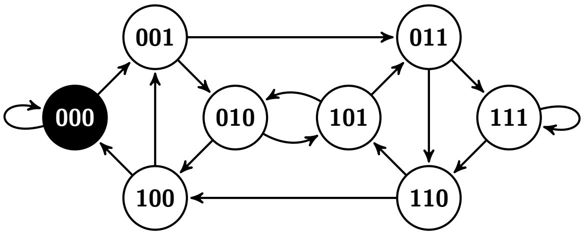

Figure 4: A minimum size determining set for $\vec{\mathcal{B}}(2,3)$ (black vertex).

is a determining set. However, though domination is preserved by automorphisms, vertices are not necessarily uniquely identifiable by their relationship to a dominating set. Thus a dominating set is not necessarily a determining set. However, the relationships above mean that the size of a minimum determining set must be at most the size of a minimum identifying code or the directed metric dimension. For de Bruijn graphs we have shown that the latter numbers are rather large. Does this mean that the determining number is also large. We will see in Corollary 4.12 that the answer for directed de Bruijn graphs is a resounding 'No'.

Lemma 4.9. *$S = \{00, 11, 22, 33, \ldots, (d-1)(d-1)\}$ is a determining set for $\vec{\mathcal{B}}(d, 2)$.*

Proof. Suppose that $\sigma \in \mathrm{Aut}(\vec{\mathcal{B}}(d, 2))$ fixes S pointwise. That is, $\sigma(ii) = ii$ for all $i \in \mathcal{A}$. Choose $ij \neq rs \in V(\vec{\mathcal{B}}(d, 2))$. Then either $i \neq r$ or $j \neq s$ (or both). If $i \neq r$ then $\vec{d}(ii, rs) = 2$ which is distinct from $\vec{d}(ii, ij) = 1$. Since an automorphism of a directed graph must preserve directed distance, $\sigma(ij) \neq rs$ if $i \neq r$. If $j \neq s$, then $\vec{d}(rs, jj) = 2$ which is distinct from $\vec{d}(ij, jj) = 1$. Thus, again using that σ preserves directed distance, $\sigma(ij) \neq rs$ if $j \neq s$. Thus, $\sigma(ij) = ij$ for all $ij \in V(\vec{\mathcal{B}}(d, 2))$ and therefore σ is the identity map and S is a determining set. □

Note that we are using directed distances both from and to elements of the set S. Thus S does not fit the definition of a directed resolving set for $\vec{\mathcal{B}}(d, 2)$ (by [6], this would require that each vertex $v \in V(G)$ be distinguished by it directed distance to the vertices of the resolving set). However directed distances both to and from a set can be used in determining automorphisms of a directed graph.

Lemma 4.10. $\mathrm{Aut}(\vec{\mathcal{B}}(d,2)) \cong \mathrm{Sym}(\mathcal{A}_d)$.

Proof. Let $\sigma \in \mathrm{Sym}(\mathcal{A}_d)$. Define φ_σ on $V(\vec{\mathcal{B}}(d,n))$ by applying σ to each vertex coordinate-wise. That is $\varphi_\sigma(ab) = \sigma(a)\sigma(b)$. It is easy to show that φ_σ preserves directed edges and thus is an automorphism. Further, distinct permutations in $\mathrm{Sym}(\mathcal{A}_d)$ produce distinct automorphisms since they act differently on the vertices of the determining set S defined above. Thus we have an injection $\mathrm{Sym}(\mathcal{A}_d) \hookrightarrow \mathrm{Aut}(\vec{\mathcal{B}}(d,2))$.

Since the vertices of S are precisely the vertices with loops, every automorphism of $\vec{\mathcal{B}}(d,2)$, must preserve S setwise. This provides the necessary injection from $\mathrm{Aut}(\vec{\mathcal{B}}(d,2)) \hookrightarrow \mathrm{Sym}(\mathcal{A})$. Thus, $\mathrm{Aut}(\vec{\mathcal{B}}(d,2)) \cong \mathrm{Sym}(\mathcal{A}_d)$. □

Note that we can consider the automorphisms of $\vec{\mathcal{B}}(d,2)$ as permutations of the loops, but we can simultaneously consider them as permutations of the symbols in the alphabet $\mathcal{A}_d$. It can be useful to view the automorphisms in these two different ways.

Note that as shown in [1], $\vec{\mathcal{B}}(d,n)$ can be built inductively from $\vec{\mathcal{B}}(d,n-1)$ in the following way. The vertex $x_1 \dots x_n \in V(\vec{\mathcal{B}}(d,n))$ corresponds to the directed edge from $x_1 \dots x_{n-1}$ to $x_2 \dots x_n$ in $\vec{\mathcal{B}}(d,n-1)$. The directed edge $\vec{\mathcal{B}}(d,n)$ from $x_1 \dots x_n \to x_2 \dots x_n x_{n+1}$ corresponds to the directed 2-path $x_1 \dots x_{n-1} \to x_2 \dots x_n \to x_3 \dots x_{n+1}$ in $\vec{\mathcal{B}}(d,n-1)$. That is, $\vec{\mathcal{B}}(d,n)$ is the directed line graph of the directed graph $\vec{\mathcal{B}}(d,n-1)$. Thus, by [7] (Chapter 27, Section 1.1), $\mathrm{Aut}(\vec{\mathcal{B}}(d,n-1)) = \mathrm{Aut}(\vec{\mathcal{B}}(d,n)) \cong \mathrm{Sym}(\mathcal{A})$. In the following paragraphs we see detail this correspondence.

Suppose that $\varphi \in \mathrm{Aut}(\vec{\mathcal{B}}(d,n))$. Since φ preserves directed edges, we know that both $\varphi(x_1 \dots x_n) = a_1 \dots a_n$ and $\varphi(x_2 \dots x_{n+1}) = b_1 \dots b_n$ if and only if $a_2 = b_1, \dots, a_n = b_{n-1}$. Thus if $\varphi(x_1 \dots x_{n-1}x_n) = a_1 \dots a_{n-1}a_n$ then for every $b \in \mathcal{A}$, $\varphi(x_1 \dots x_{n-1}z) = a_1 \dots a_{n-1}c$ for some $c \in \mathcal{A}$. In particular, this allow us to define an automorphism $\varphi' \in \mathrm{Aut}(\vec{\mathcal{B}}(d,n-1))$ corresponding to $\varphi \in \mathrm{Aut}(\vec{\mathcal{B}}(d,n))$. Define φ' by $\varphi'(x_1 \dots x_{n-1}) = a_1 \dots a_{n-1}$ where $\varphi(x_1 \dots x_n) = a_1 \dots a_{n-1}$. By the preceding discussion, φ' is well-defined. It is also clearly a bijection on vertices of $\vec{\mathcal{B}}(d,n-1)$. Consider $x_1 \dots x_{n-1}$ and $x_2 \dots x_{n-1}x_n$, the initial and terminal vertices of a directed edge in $\vec{\mathcal{B}}(d,n-1)$. Since φ preserves directed edges if $\varphi(x_1 \dots x_n) = a_1 \dots a_{n-1}a_n$ then for any $z \in \mathcal{A}$, $\varphi(x_2 \dots x_n z) = a_2 \dots a_n w$ for some $w \in \mathcal{A}$. By definition of φ',$\varphi'(x_1, \dots x_{n-1}) = a_1 \dots a_{n-1}$ and $\varphi'(x_2 \dots x_n) = a_2 \dots a_n$. Thus φ' preserves the directed edge. Thus we get $\mathrm{Aut}(\vec{\mathcal{B}}(d,n)) \hookrightarrow \mathrm{Aut}(\vec{\mathcal{B}}(d,n-1))$.

In the other direction, suppose we are given $\varphi' \in \text{Aut}(\vec{\mathcal{B}}(d, n-1))$. Since φ' preserves directed edges, and directed edges of $\vec{\mathcal{B}}(d, n-1)$ are precisely the vertices of $\vec{\mathcal{B}}(d, n)$, φ' defines a map on vertices of $\vec{\mathcal{B}}(d, n)$. That is, (with some abuse of notation)

$$\begin{aligned}
\varphi(x_1 \ldots x_n) &= \varphi(x_1 \ldots x_{n-1} \to x_2 \ldots x_n) \\
&= \varphi'(x_1 \ldots x_{n-1} \to x_2 \ldots x_n) \\
&= \varphi'(x_1 \ldots x_{n-1}) \to \varphi'(x_2 \ldots x_n).
\end{aligned}$$

Thus, given $\varphi'(x_1 \ldots x_{n-1}) = a_1 \ldots a_{n-1}$ then $\varphi'(x_2 \ldots x_n) = a_2 \ldots a_n$ for some $a_n \in \mathcal{A}$ and we define $\varphi(x_1 \ldots x_n) = a_1 \ldots a_n$. Further, since φ' preserves directed 2-paths, φ preserves directed edges. Thus we get

$$\text{Aut}(\vec{\mathcal{B}}(d, n-1)) \hookrightarrow \text{Aut}(\vec{\mathcal{B}}(d, n)).$$

Since the automorphisms of $\vec{\mathcal{B}}(d, 2)$ are permutations of the loops, and of the symbols of $\mathcal{A}$, by induction, so are the automorphisms of $\vec{\mathcal{B}}(d, n)$ for all n. Thus we have proved the following.

Theorem 4.11. $\text{Aut}(\vec{\mathcal{B}}(d, n)) \cong \text{Sym}(\mathcal{A}_d)$ *for all* $n \geq 2$.

Corollary 4.12. $\text{Det}(\vec{\mathcal{B}}(d, n)) = \lceil \frac{d-1}{n} \rceil$.

Proof. Let S be a minimum set of vertices in which each letter of $\mathcal{A}_{d-1}$ occurs at least once. It is easy to see that $|S| = \lceil \frac{d-1}{n} \rceil$. Any permutation of $\mathcal{A}_d$ that acts nontrivially on any letter of $\mathcal{A}_d$ must act non-trivially on any string containing that letter. Thus if $\sigma \in \text{PtStab}(S)$, then σ must fix every letter contained in any string in S. Thus σ fixes $0, 1, \ldots, d-1$ and therefore also d. We can conclude that σ is the identity in both $\text{Sym}(\mathcal{A}_d)$ and in $\text{Aut}(\vec{\mathcal{B}}(d, n)$ and therefore S is a determining set. Thus $\text{Det}(\vec{\mathcal{B}}(d, n)) \leq \lceil \frac{d-1}{n} \rceil$.

Further if $|S| < \lceil \frac{d-1}{n} \rceil$ then fewer than $d-1$ letters of $\mathcal{A}_d$ are used in strings in S. If $a, b \in A_d$ are not represented in S, then the transposition $(a\ b)$ in $\text{Sym}(A_d)$ is a non-trivial automorphism of $\vec{\mathcal{B}}(d, n)$ that fixes S pointwise. Thus S is not a determining set. □

Thus for directed de Bruijn graphs, the determining number and the directed metric dimension can be vastly different in size.

5 Future Work

There are several directions that future work in this research area could take. The first is to continue the research on identifying codes in de Bruijn graphs. One key result missing from this paper is the determination of the size of a minimum 1-identifying code in the graph $\vec{\mathcal{B}}(2,n)$ for n even. Our results show that this size is at most $2^{n-1}+1$, and at least 2^{n-1}, leaving little room between the two bounds. Additionally, constructions are still needed for t-identifying codes in $\vec{\mathcal{B}}(d,n)$ for $t>3$. As we have shown that t-identifying codes do not always exist in directed de Bruijn graphs, it would be ideal to determine both a formula for which graphs $\vec{\mathcal{B}}(d,n)$ are t-identifiable for a given t, as well as constructions when it is known that such an identifying code exists.

An alternative direction for future research is to consider these same vertex subsets (identifying codes, dominating sets, resolving sets, and determining sets) on the undirected de Bruijn graph. Little work has been done and even foundational results like the size of a minimum dominating set are currently missing. Basic Matlab programs have shown that many more undirected de Bruijn graphs are t-identifiable than directed, and also that the minimum size of an identifying code is much smaller. For example, through brute force testing we have determined that the minimum size of a 1-identifying code in $\mathcal{B}(2,5)$ is 12, whereas in the directed graph $\vec{\mathcal{B}}(2,5)$ we have shown that the minimum size is 16.

Finally, variations on the concept of identifying code would be useful for real-world applications. For example, one type of variation known as a k-robust identifying code allows for up to k sensor (identifying code vertex) failures without disruption of the identifying code properties.

References

[1] Toru Araki. On the k-tuple domination of de bruijn and kautz digraphs. *Inform. Process. Lett.*, 104(3):86–90, 2007. 4.3

[2] Joel Baker. De bruijn graphs and their applications to fault tolerant networks. Master's thesis, Oregon State University, 2011. 1

[3] Debra L. Boutin. Identifying graph automorphisms using determining sets. *Electron. J. Combin.*, 13(1):R78, 2006. 4.8

[4] Irène Charon, Iiro Honkala, Olivier Hudry, and Antoine Lobstein. Structural properties of twin-free graphs. *Electron. J. of Combin.*, 14(1):R16, 2007. 3.13

[5] Irène Charon, Olivier Hudry, and Antoine Lobstein. Minimizing the size of an identifying or locating-dominating code in a graph is NP-hard. *Theoret. Comput. Sci.*, 290(3):2109–2120, 2003. 1

[6] Gary Chartrand, Michael Raines, and Ping Zhang. The directed distance dimension of oriented graphs. *Math. Bohem.*, 125(2):155–168, 2000. 4.2, 4.3

[7] R. L. Graham, M. Grötschel, and L. Lovász, editors. *Handbook of combinatorics. Vol. 1, 2.* Elsevier Science B.V., Amsterdam; MIT Press, Cambridge, MA, 1995. 4.3

[8] Mark G. Karpovsky, Krishnendu Chakrabarty, and Lev B. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE Trans. Inform. Theory*, 44(2):599–611, 1998. 1

[9] Samir Khuller, Balaji Raghavachari, and Azriel Rosenfeld. Landmarks in graphs. *Discrete Appl. Math.*, 70:217–229, 1996. 1

[10] Y. Kikuchi and Y. Shibata. On the domination numbers of generalized de bruijn digraphs and generalized kautz digraphs. *Inform. Process. Lett.*, 86:79–85, 2003. 4.2, 4.1

[11] Andréa Richa, Christian Scheideler, and Phillip Stevens. Self-stabilizing de bruijn networks. In *Stabilization, Safety, and Security of Distributed Systems*, pages 416–430. Springer, 2011. 1

[12] Steven S. Skiena. *The Algorithm Design Manual.* Springer Publishing Company, Incorporated, 2nd edition, 2008. 1