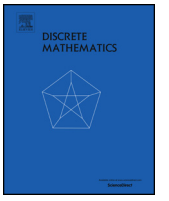
Discrete Mathematics 346 (2023) 113310

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc



Levenshtein graphs: Resolvability, automorphisms & determining sets

Perrin E. Ruth, Manuel E. Lladser *

Department of Applied Mathematics, University of Colorado, Boulder, United States of America

ARTICLE INFO

Article history:
Received 26 July 2021
Received in revised form 18 February 2022
Accepted 23 December 2022
Available online 13 January 2023

Keywords:
Edit distance
Hamming graph
Levenshtein graph
Multilateration
Node2vec
Resolving set

ABSTRACT

We introduce the notion of Levenshtein graphs, an analog to Hamming graphs but using the edit distance instead of the Hamming distance; in particular, vertices in Levenshtein graphs may be strings (i.e., words or sequences of characters in a reference alphabet) of possibly different lengths. We study various properties of these graphs, including a necessary and sufficient condition for their shortest path distance to be identical to the edit distance, and characterize their automorphism group and determining number. We also bound the metric dimension (i.e. minimum resolving set size) of Levenshtein graphs. Regarding the latter, recall that a run is a string composed of identical characters. We construct a resolving set of two-run strings and an algorithm that computes the edit distance between a string of length k and any single-run or two-run string in $O(k)$ operations.

1. Introduction

For a general unweighted graph $G = (V, E)$, a set $R \subset V$ is called resolving when for all $u, v \in V$, if $d(u, r) = d(v, r)$ for each $r \in R$ then $u = v$. Here and in what follows, $d(\cdot, \cdot)$ denotes the (graph) distance, i.e., shortest path distance, between pairs of vertices in the corresponding graph. $\beta(G)$, the metric dimension of G, is defined as the size of a smallest possible resolving set of G [23,11]. The problem of finding the metric dimension of an arbitrary graph is NP-Complete [6,9,14]. Nevertheless, when the distance matrix of a graph can be computed explicitly, resolving sets of size $\big(1 + \{1 + o(1)\} \cdot \ln|V|\big) \cdot \beta(G)$ may be found using the so-called Information Content Heuristic (ICH) [12]. For a concise exposition of metric dimension see [25], and for a detailed exposition see [26].

If $R = \{r_1, \ldots, r_n\} \subset V$ of cardinality n resolves G, then the transformation

$$d(v|R) := \big(d(v, r_1), \ldots, d(v, r_n)\big) \tag{1}$$

from V into $\mathbb{R}^n$ represents nodes in G as n-dimensional vectors in a one-to-one manner. Furthermore, since for each $u, v \in V$ and $r \in R$, $|d(u, r) - d(v, r)| \leq d(u, v)$, $d(\cdot|R)$ maps nearby nodes in G into tuples with similar coordinates in $\mathbb{R}^n$. These two properties are very appealing to represent nodes in G as Euclidean vectors—offering an alternative to other graph embedding techniques such as node2vec [10]. To fix ideas, in the context of network science, graph distance is often a relevant feature in the community recovery problem. Here, nodes in a graph are assumed to be partitioned into disjoint but unknown subsets called communities, which influence how edges are placed between the nodes. Resolving set based

* Corresponding author.
E-mail address: manuel.lladser@colorado.edu (M.E. Lladser).

https://doi.org/10.1016/j.disc.2022.113310

embeddings induce a natural numerical representation (i.e. feature vector) for each node on which to base the community recovery. Of course, the smaller the cardinality of a resolving set, the smaller the dimension of the embedding, which motivates the study of metric dimension, and of algorithms capable of efficiently finding small resolving sets.

The Hamming distance between two strings u and v of the same length, denoted as $h(u,v)$, is the total number of mismatches between u and v. (The length of a string w is denoted $|w|$.) Up to a graph isomorphism, the Hamming graph $\mathbb{H}_{k,a}$, with $k,a \geq 1$ integers, has as vertices all strings of length k formed using the characters in $\{0,\ldots,a-1\}$, and two vertices u and v are neighbors if and only if $h(u,v)=1$. As a result, the distance between nodes in $\mathbb{H}_{k,a}$ is precisely their Hamming distance; in particular, Hamming graphs are connected. We call k the dimension and a the alphabet size of $\mathbb{H}_{k,a}$, respectively.

Much is known already about Hamming graphs, including their automorphism group [5] and their asymptotic metric dimension. Indeed [13]:

$$\beta(\mathbb{H}_{k,a}) \sim \frac{2k}{\log_a(k)}, \text{ as } k \to \infty,$$

and because the proof of this result is constructive, a resolving set of $\mathbb{H}_{k,a}$ of approximate relative size $2k/\log_a(k)$ may be found for k large enough. Otherwise, starting from a resolving set of $H_{k-r,a}$ of some size s (e.g., obtained using the ICH), a resolving set for $\mathbb{H}_{k,a}$ of size $s + r\lfloor a/2 \rfloor$ may be found recursively in $O(ar^2)$ time [27]. Recent work has shown how to identify unnecessary nodes in a resolving set [15]; which may provide better non-asymptotic estimates for $\beta(\mathbb{H}_{k,a})$.

As mentioned earlier, resolving sets of graphs are useful to represent their nodes as Euclidean vectors by means of transformations such as in equation (1). In particular, resolving sets in Hamming graphs may be used to represent symbolic sequences (e.g., words and genomic sequences) numerically. Unfortunately, this capability is limited to sequences of the same length, and a chief motivation of this paper is to overcome this equal length limitation.

The edit distance—also called the Levenshtein distance [16]—between two strings u and v of possibly different lengths is defined as the minimal number of character substitutions, deletions, or insertions required to transform one string into the other. We denote this quantity as $\ell(u,v)$. Since the Hamming distance can be thought of as the minimal number of substitutions to transform one string into the other, if $|u| = |v|$ then $\ell(u,v) \leq h(u,v)$.

The edit distance can be computed using so-called alignments. To explain how, consider two non-empty strings u and v of possibly different lengths, and let $\mathcal{S}$ be the set of symbols (i.e., characters) forming the strings. Let - denote a symbol outside $\mathcal{S}$, from now on called a gap. A gap conveys either a character insertion in one of the strings or a character deletion in the other.

An alignment between u and v is a pair of strings $u' = u'_1 \cdots u'_k$ and $v' = v'_1 \cdots v'_k$ of some same length $k \geq 1$, formed using characters in $\mathcal{S} \cup \{\text{-}\}$, such that (i) u and v occur as possibly non-contiguous sub-strings of u' and v', respectively; and (ii) there is no $1 \leq i \leq k$ such that $u'_i = v'_i = \text{-}$. Alignments are visualized placing u' and v' in a two-dimensional array so that for each $1 \leq i \leq k$, the i-th character of u' is aligned on top of the i-th character of v'. When $u'_i, v'_i \in \mathcal{S}$, we say that there is a match at position i if $u'_i = v'_i$, and a mismatch if $u'_i \neq v'_i$. Otherwise, if $u'_i = \text{-}$ or $v'_i = \text{-}$, we say that there is a gap at that position. Recall that no gap may be placed on top of another one in an alignment.

The score of the alignment is defined as $\sum_{i=1}^{k} [\![u'_i \neq v'_i]\!]$, where $[\![\cdot]\!]$ is our notation for indicator functions. Namely, $[\![\cdot]\!]$ takes the value 1 if the statement within the double-bracket parentheses is true otherwise is 0. The edit distance between two non-empty strings corresponds to the lowest score among all possible alignments of the strings [7]. Any such alignment is called optimal. To fix ideas, equations (2)–(4) display three different alignments between the strings 001 and 01. The score of the alignment A is two because the second 0 in the first row is mismatched with the character 1 in the second row, and the 1 in the first row is aligned against a gap. Similarly, the scores of alignments B and C are one. Since the score of any alignment between different strings must be one or larger, it follows that $\ell(001,01) = 1$, and B and C are optimal alignments of 001 and 01.

$$A := \begin{matrix} 0 & 0 & 1 \\ 0 & 1 & \text{-} \end{matrix}; \tag{2}$$

$$B := \begin{matrix} 0 & 0 & 1 \\ 0 & \text{-} & 1 \end{matrix}; \tag{3}$$

$$C := \begin{matrix} 0 & 0 & 1 \\ \text{-} & 0 & 1 \end{matrix}. \tag{4}$$

Optimal alignments can be determined and scored through a well-known dynamic programming approach, which has been invented many times in different contexts [16,19,30]. For strings $u = u_1 \ldots u_m$ and $v = v_1 \ldots v_n$ of lengths m and n, respectively, where u_i and v_j denote alphabet characters, this algorithm computes the columns (or rows) of the $m \times n$ matrix with entries $d_{i,j} := \ell(u_1 \ldots u_i, v_1 \ldots v_j)$ via the recursion:

$$d_{i,j} = \min\left\{d_{i-1,j-1} + [\![u_i \neq v_j]\!], d_{i-1,j} + 1, d_{i,j-1} + 1\right\}. \tag{5}$$

The time complexity of this algorithm is $O(mn)$, which is expensive for long pairs of strings; however, by focusing on the diagonals of the matrix $(d_{i,j})$, as oppose to its columns or rows, it is possible to speed up the calculations to an $O\big(\ell(u,v) \cdot \min\{m,n\}\big)$ complexity [28].

Preliminaries and related work. To overcome the length limitation of Hamming graphs, we adopt the following definition.

Definition 1.1. For integers $0 \leq k_1 \leq k_2$ and $a \geq 2$, the Levenshtein graph $\mathbb{L}_{k_1,k_2;a}$ has as vertices all strings of a length between k_1 and k_2 (inclusive) formed using the characters in $\{0,\ldots,a-1\}$, and two nodes u and v are connected by an edge iff $\ell(u,v)=1$. We denote the vertex and edge set of this graph as $V_{k_1,k_2;a}$ and $E_{k_1,k_2;a}$, respectively. (See Fig. 1.)

Fig. 1. Visual representation of $\mathbb{L}_{0,1;3}$ (left), and $\mathbb{L}_{3,3;2}$ (right).

Unless otherwise stated, it is assumed in what follows that $0 \leq k_1 \leq k_2$ and $a \geq 2$.

Observe that, for $k_1 \leq k \leq k_2$, the subgraph of nodes in $\mathbb{L}_{k_1,k_2;a}$ of length k is precisely $\mathbb{H}_{k,a}$. Further, only nodes of equal or consecutive length can be neighbors in $\mathbb{L}_{k_1,k_2;a}$ (see Fig. 2).

Ahead we write $\mathbb{L}_{k;a}$ as shorthand for $\mathbb{L}_{0,k;a}$. Accordingly, we denote the vertex and edge set of $\mathbb{L}_{k;a}$ as $V_{k;a}$ and $E_{k;a}$, respectively. The empty string, denoted as ϵ, is the only vertex of length zero in this graph. Besides, we define $\mathbb{L}_a$ as the graph with vertex set $\cup_{k\geq 1} V_{k;a}$ where two nodes u and v of arbitrary length are neighbors if and only if $\ell(u,v)=1$. All nodes in $\mathbb{L}_a$ have finite length.

Various other notions of Levenshtein graphs have been considered in the literature, usually motivated by specific applications. One common definition is that two nodes are neighbors when their edit distance is underneath some threshold. For instance, Pisanti [20] defines Levenshtein graphs over a vertex set of arbitrary genes, and two genes u and v are joined by an edge when $\ell(u,v) \leq t$; which they use to test random graphs as viable models for genomic data.

Sala et al. [22] define the vertex set of Levenshtein graphs as $\{0,\ldots,a-1\}^k$, and two nodes u and v are declared neighbors when they may be aligned using at most $2t$ gaps (alternatively, u and v are said to have a fixed-length Levenshtein distance of t [2]). They use this construction to find the maximal number of common supersequences between two strings of the same length and the maximal number of subsequences of a given string. This is motivated by error correcting codes on the insertion/deletion channel—the subject of Levenshtein's seminal paper [16]. Motivated by the same problem, Bar-Lev, Etzion, and Yaakobi [2] address the specific case with $t=1$ to study the size of balls of radius one on these graphs.

Zhong, Heinicke, and Rayner [31] define the vertex set of the Levenshtein graph to have nodes corresponding to microRNAs in mice and people, and u and v are connected by an edge only when $\ell(u,v) \leq 3$.

Finally, Stahlberg [24] defines the vertex set of Levenshtein graphs from all strings of a given set M as well as all strings that lie on a shortest path between two strings in M, and nodes u and v are then joined by an edge if and only if $\ell(u,v)=1$.

Since $\mathbb{L}_{k,k;a}$ is isomorphic to $\mathbb{H}_{k,a}$; Levenshtein graphs include Hamming graphs as special cases. Nevertheless, as pointed out in [29], which implicitly uses a notion similar to ours, Levenshtein graphs cannot be represented as Cartesian products when $k_1 < k_2$. This makes their study particularly challenging.

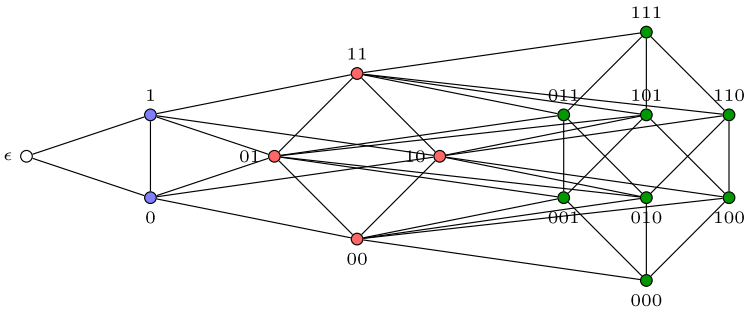


Fig. 2. Visualization of $\mathbb{L}_{3;2}$. The sub-graphs of all strings of fixed length are Hamming graphs: the white, blue, red, and green nodes form $\mathbb{H}_{0,2}$, $\mathbb{H}_{1,2}$, $\mathbb{H}_{2,2}$, and $\mathbb{H}_{3,2}$, respectively. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Paper organization. In this manuscript we initiate a study of Levenshtein graphs—as given in Definition 1.1. The manuscript is based on the recent Honors Thesis by the first author [21].

In Section 2, we show that Levenshtein graphs are always connected, and provide a necessary and sufficient condition for their distance to coincide with the edit distance between all pairs of nodes. Unlike Hamming graphs, the edit and graph distance between all pairs of nodes in a Levenshtein graph is not necessarily the same. For instance, in $\mathbb{L}_{3,3;2}$, $\ell(010, 101) = 2$ but $d(010, 101) = 3$ (see Fig. 1). Nevertheless, in $\mathbb{L}_{0,3;2}$, $d(010, 101) = 2$ (see Fig. 2).

In Section 3, we show a formula to describe the edit distance of an arbitrary string to a string with at most two runs (a run is a maximal substring of a single repeated character in a string). This formula leads to an algorithm to compute the distance from any string u to any string with at most two runs in $O(|u|)$ time, which is faster than many common methods of computing the edit distance. The results in sections 4-5 rely heavily on Section 3. In Section 4, we construct a resolving set of $\mathbb{L}_{k_1,k_2;a}$ of size $O(ak_2(k_2 - k_1 + 1))$ explicitly. Since nodes on this set have at most two runs, we may utilize the algorithm from Section 3 to multilaterate efficiently any string of length between k_1 and k_2.

In Section 5, we characterize the automorphism group of Levenshtein graphs, which has fixed size $2a!$ when $k_1 < k_2$ and $k_2 \geq 2$. Finally, in Section 6, we address the determining number [3,8] of Levenshtein graphs. This notion is useful for describing graph automorphisms. For a given graph $G = (V, E)$, a set $S \subset V$ is called determining if whenever f and g are automorphisms of G such that $f(s) = g(s)$, for all $s \in S$, then $f = g$. The determining number of a graph is the size of its smallest determining set. For $k_1 < k_2$ with $k_2 \geq 2$ and $(k_2, a) \neq (2, 2)$, we show that the determining number of $\mathbb{L}_{k_1,k_2;a}$ is $\lceil a/k_2 \rceil$.

2. Graph versus edit distance, and connectivity

The distance between pairs of nodes in a Hamming graph is equal to their Hamming distance; however, as already pointed out in the Introduction, this is not necessarily the case for Levenshtein graphs. The main result in this section is the following one.

Theorem 2.1. *Levenshtein graphs are connected, and the distance between every pair of nodes on $\mathbb{L}_{k_1,k_2;a}$ is equal to their edit distance if and only if $k_1 < k_2$ or $k_1 = k_2 \leq 2$. If $k > 2$ then the graph distance in $\mathbb{L}_{k,k;a}$ is the Hamming distance.*

This theorem is a direct consequence of the following three lemmas.

Ahead, the length of a path is understood as the number edges that compose it. In addition, $w_{(n)}$ and $w^{(n)}$ denote the prefix and suffix of length n of a word w, respectively.

Lemma 2.1. Let $k_1 < k_2$. For all nodes u and v in $\mathbb{L}_{k_1,k_2;a}$, there is a path of length $\ell(u, v)$ that connects u with v. In particular, $\mathbb{L}_{k_1,k_2;a}$ is connected, and for all $u, v \in V_{k_1,k_2;a}$, $d(u, v) \leq \ell(u, v)$.

Proof. We show something more general, namely, for any alignment between two nodes in a Levenshtein graph there is a path of the same length as the alignment score that connects them, while visiting only nodes of a length between the shortest and longest of the two. This suffices to prove the lemma because if A is an optimal alignment between u and v, and p a path in $\mathbb{L}_{k_1,k_2;a}$ of length score(A), then $d(u, v) \leq \text{length}(p) = \text{score}(A) = \ell(u, v)$.

Consider any alignment A between two nodes u and v. Define $\delta := |u| - |v|$. Since alignment scores are invariant under permutations of their rows, as well as their columns, we may assume without any loss of generality that $|u| \geq |v|$, and that A is of the form:

$$A = \begin{array}{c|c|c|c} u_0 & u_1 & u_2 & \text{-}^k \\ v_0 & \text{-}^\delta & \text{-}^k & v_2 \end{array};$$

where the u_i's and v_i's are nodes in $\mathbb{L}_{k_1,k_2;a}$ such that $|u_0| = |v_0| \geq 0$, $|u_1| = \delta$, $|u_2| = |v_2| = k$ for some $k \geq 0$, and -^n denotes n consecutive gaps.

Let s_0 denote the score of the alignment associated with u_0 and v_0 above. Clearly, we can construct a path of length s_0 from $u = u_0u_1u_2$ to $v_0u_1u_2$ substituting, one at a time, the mismatched characters in u_0 by the corresponding characters in v_0. Since substitutions do not alter the length of a node, all nodes in this path have length $|u|$.

Next, we can construct a path of length δ from $v_0u_1u_2$ to v_0u_2 deleting, one at a time, the characters in u_1. In particular, the nodes in this path have a (decreasing) length between $|v_0u_1u_2| = |u|$ and $|v_0u_2| = |v|$, inclusive.

We can now construct a path of length $2k$ from v_0u_2 to $v_0v_2 = v$, stitching the following paths of length 2. When $|v| < k_2$, each of these paths is obtained by inserting a character from v_2, and subsequently deleting another in u_2. As a result, all nodes in these paths have a length between $|v|$ and $|v| + 1 \leq k_2$, inclusive. The short paths are:

$$v_0\, u_2^{(k)}\, v_{2(0)},\ v_0\, u_2^{(k-1)}\, v_{2(0)},\ v_0\, u_2^{(k-1)}\, v_{2(1)};$$
$$v_0\, u_2^{(k-1)}\, v_{2(1)},\ v_0\, u_2^{(k-2)}\, v_{2(1)},\ v_0\, u_2^{(k-2)}\, v_{2(2)};$$
$$\vdots$$

$$v_0\, u_2^{(1)}\, v_{2(k-1)},\ v_0\, u_2^{(0)}\, v_{2(k-1)},\ v_0\, u_2^{(0)}\, v_{2(k)}.$$

Similarly, when $|v| = k_2$, each of these paths is obtained by deleting a character in v_2, and subsequently inserting a character from u_2. All nodes in these paths have a length between $|v|$ and $|v| - 1 \geq k_1$ inclusive.

Appending all the previous paths, we obtain a path from u to v of length $s_0 + \delta + 2k$, which is precisely the score of A. Since each node in this path is contained in $\mathbb{L}_{k_1,k_2;a}$, the lemma follows. □

Lemma 2.2. Let $k_1 < k_2$. For all nodes u and v in $\mathbb{L}_{k_1,k_2;a}$, $d(u, v) \geq \ell(u, v)$.

Proof. Clearly, $d(u, v) = 0$ if and only if $\ell(u, v) = 0$. Thus, without loss of generality, we may assume that $n := d(u, v) \geq 1$. Due to Lemma 2.1, $\mathbb{L}_{k_1,k_2;a}$ is connected and hence n is finite. In particular, there is in $\mathbb{L}_{k_1,k_2;a}$ a (simple) path $w_0 = u, \ldots, w_n = v$ of length n that connects u and v. Since $d(w_i, w_{i+1}) = \ell(w_i, w_{i+1}) = 1$, the triangular inequality implies that:

$$d(u, v) = \sum_{i=0}^{n-1} d(w_i, w_{i+1}) = \sum_{i=0}^{n-1} \ell(w_i, w_{i+1}) \geq \ell(u, v),$$

which shows the lemma. □

Lemma 2.3. For all $k \geq 0$, $\mathbb{L}_{k,k;a} = \mathbb{H}_{k;a}$; in particular, $\mathbb{L}_{k,k;a}$ is connected. Further, the distance between every pair of nodes on $\mathbb{L}_{k,k;a}$ is equal to their edit distance if and only if $k \leq 2$.

Proof. To show the first claim, it suffices to show that $\mathbb{L}_{k,k;a}$ and $\mathbb{H}_{k,a}$ have the same edges. Indeed, if $h(u, v) = 1$ then u and v can be aligned perfectly except for one mismatch. In particular, $\ell(u, v) \leq 1$. But, since $u \neq v$, $\ell(u, v) > 0$, hence $\ell(u, v) = 1$. Conversely, if $\ell(u, v) = 1$ then an optimal alignment between u and v consists of a single mismatch, or a single gap. Since the latter is not possible because $|u| = |v|$, $h(u, v) = 1$, which shows the claim.

Due to the first claim, $d(u, v) = h(u, v)$ for all pair of nodes u, v in $\mathbb{L}_{k,k;a}$. We use this to show the second claim, assuming, without loss of generality, that $u \neq v$.

The second claim is trivial when $k = 0$. If $k = 1$ then, as we argued before, $\ell(u, v) = 1 = h(u, v) = d(u, v)$. Instead, if $k = 2$ and $h(u, v) = 1$ then, as we just argued, $\ell(u, v) = 1 = h(u, v) = d(u, v)$. Otherwise, if $k = 2$ but $h(u, v) = 2$ then Lemma 2.2 implies that $0 < \ell(u, v) \leq 2$; however, $\ell(u, v) = 1$ is not possible because the optimal alignment between u and v would then have to use a single gap, which in turn is not possible because u and v are of the same length. Hence, $\ell(u, v) = 2$ and again $\ell(u, v) = h(u, v) = d(u, v)$.

Finally, if $k > 2$, and since $a \geq 2$, there is in $\mathbb{L}_{k,k;a}$ a node u of length k formed by alternating 0's and 1's. Let v be the flip of u. Then $h(u, v) = k$ but $\ell(u, v) \leq 2$ because the strings $-u$ and $v-$ align perfectly except for their ends; in particular, $h(u, v) > \ell(u, v)$ i.e. $d(u, v) > \ell(u, v)$. □

3. Edit distance to a string with at most two runs

In this section, we obtain rather explicit formulas for the edit distance between an arbitrary string and another one with at most two runs. These will prove useful for studying the resolvability of Levenshtein graphs and their automorphism group.

In what follows the total number of occurrences of an alphabet character α in a string w is denoted $N_\alpha(w)$, whereas the number of runs in w is denoted $r(w)$. For example, $N_0(01121) = 1$, $N_1(01121) = 3$, $N_2(01121) = 1$, and $r(01121) = 4$.

The main result in this section is the following.

Theorem 3.1. *Let $l, r \geq 0$ be integers and α, β different alphabet characters. Then, for any string w:*

$$\ell(w, \alpha^l) = \max\{|w|, l\} - \min\{N_\alpha(w), l\}; \tag{6}$$

$$\ell(w, \alpha^l\beta^r) = \min_{i_0 \leq i \leq i_1} \ell\big(w_{(i)}, \alpha^l\big) + \ell\big(w^{(|w|-i)}, \beta^r\big); \tag{7}$$

where $i_0 := \max\{0, \min\{l, |w| - r\}\}$ and $i_1 := \min\{|w|, \max\{l, |w| - r\}\}$.

A noteworthy consequence of this theorem is the following.

Corollary 3.1. *If u and v are strings such that $|u| = |v|$, and u or v have at most two runs, then $\ell(u, v) = h(u, v)$.*

Proof. Suppose that $|u| = |v| = k$, and write $u = u_1 \cdots u_k$ with $u_1, \ldots, u_k$ alphabet characters. Without any loss of generality assume that $r(v) \leq 2$.

If $r(v) = 0$ then $u = v$; in particular, $\ell(u, v) = 0 = h(u, v)$. Instead, if $r(v) = 1$ then $v = \alpha^k$ for some alphabet character α, and Equation (6) implies that

$$\ell(u, v) = k - N_\alpha(u) = \sum_{i=1}^{k} [\![u_i \neq \alpha]\!] = h(u, v).$$

Finally, if $r(v) = 2$ then $v = \alpha^l \beta^{k-l}$ for some integer $1 \leq l < k$ and alphabet characters $\alpha \neq \beta$. Hence, from Equation (6), and the previous result for when $r(v) = 1$, we find that

$$\begin{aligned}\ell(u, v) &= \ell(u_1 \cdots u_l, \alpha^l) + \ell(u_{l+1} \cdots u_k, \beta^{k-l}) \\ &= h(u_1 \cdots u_l, \alpha^l) + h(u_{l+1} \cdots u_k, \beta^{k-l}) \\ &= h(u, v),\end{aligned}$$

as claimed. □

The proof of Theorem 3.1 follows from the next two results. Equation (6) is a direct consequence of Lemma 3.1, and equation (7) follows from Lemma 3.2.

Lemma 3.1. For all string w, if $l \geq 0$ and α is an alphabet character then: $\ell(w, \alpha^l) = \max\{|w|, l\} - \min\{N_\alpha(w), l\}$.

Proof. Assume that $w \neq \epsilon$ and $l > 0$, otherwise the statement is trivial. The score of an alignment is its length minus the number of matches in it. But the length of an alignment is at least the length of the longest string, and the number of matches is at most the number of characters shared by the strings. In particular, since the edit distance between w and α^l is the score of some optimal alignment, we have that: $\ell(w, \alpha^l) \geq \max\{|w|, l\} - \min\{N_\alpha(w), l\}$.

To complete the proof, it suffices to expose an alignment with the same score as the right-hand side of this inequality. For this let $n := N_\alpha(\omega)$. Assume first that α^n is a prefix of w. We now consider two cases. If $|w| \leq l$ then $w = \alpha^n u$, with $N_\alpha(u) = 0$, and the following alignment between w and α^l has the desired score:

$$\begin{array}{c|c|c} \alpha^n & u & {}_{-}{}^{l-|w|} \\ \alpha^n & \alpha^{|w|-n} & \alpha^{l-|w|} \end{array}.$$

Otherwise, if $|w| \geq l$, let $\delta = \min\{n, l\}$ and write $w = \alpha^\delta u v$, with $|u| = l - \delta$ and $|v| = |w| - l$. Now, the following alignment has the desired score:

$$\begin{array}{c|c|c} \alpha^\delta & u & v \\ \alpha^\delta & \alpha^{l-\delta} & {}_{-}{}^{|w|-l} \end{array}.$$

The previous argument assumes that α^n is a prefix of w. If this is not the case, we may shuffle the columns of the alignments to reproduce w on the top row but without altering their scores. From this, the lemma follows. □

Lemma 3.2. Let $k, l, r \geq 0$ be integers. If $w = w_1 \cdots w_k$ is a string of length k and α, β are different alphabet characters then

$$\ell(w, \alpha^l \beta^r) = \min_{i_0 \leq i \leq i_1} \ell(w_{(i)}, \alpha^l) + \ell(w^{(k-i)}, \beta^r),$$

where $i_0 := \max\{0, \min\{l, k - r\}\}$ and $i_1 := \min\{k, \max\{l, k - r\}\}$.

Proof. Without loss of generality assume that $k > 0$. Define $l_i := N_\alpha(w_{(i)})$ and $r_i := N_\beta(w^{(k-i)})$, for $0 < i < k$. Furthermore, define $l_i := 0$ and $r_i := N_\beta(w)$ for $i \leq 0$, and $l_i := N_\alpha(w)$ and $r_i := 0$ for $i \geq k$.

Any alignment A between w and $\alpha^l \beta^r$ may be segmented as

$$A = \begin{array}{c|c} u_0 & u_1 \\ v_0 & v_1 \end{array},$$

where u_0 and u_1 correspond to a possibly empty prefix and suffix of w, respectively, and v_0 and v_1 correspond to the strings α^l and β^r, respectively. (u_0, u_1, v_0, v_1 may contain gaps.) Since this also applies to an optimal alignment between w and $\alpha^l \beta^r$, it follows that

$$\begin{aligned}\ell(w, \alpha^l \beta^r) &= \min_{0 \leq i \leq k} \quad \ell(w_{(i)}, \alpha^l) + \ell(w^{(k-i)}, \beta^r) \\ &= \min_{0 \leq i \leq k} \quad \max\{l, i\} - \min\{l, l_i\} + \max\{r, k - i\} - \min\{r, r_i\} \\ &= \min_{0 \leq i \leq k} \quad \frac{k + |l - i| + |k - r - i| + |l - l_i| - l_i + |r - r_i| - r_i}{2},\end{aligned}$$

where for the second identity we have used Lemma 3.1, and for the third one the well-known identities $\max\{a,b\} = (a+b+|a-b|)/2$, and $\min\{a,b\} = (a+b-|a-b|)/2$.

Consider the functions $f_1, f_2 : \mathbb{Z} \to \mathbb{Z}$ defined as

$$f_1(i) := \frac{k-l-r}{2} + \frac{|l-i|+|k-r-i|}{2};$$
$$f_2(i) := \frac{|l-l_i|+l-l_i}{2} + \frac{|r-r_i|+r-r_i}{2}.$$

In particular, $\ell(w, \alpha^l\beta^r) = \min_{0\le i\le k} f_1(i) + f_2(i)$. Next we show that this minimum is achieved at some $i_0 \le i \le i_1$.

Observe that up to a constant summand, $f_1(i)$ is the average of the distance from i to l, and from i to $k-r$. So $f_1(i)$ is strictly decreasing for $i \le \min\{i, k-r\}$, and strictly increasing for $\max\{i, k-r\} \le i$. In particular, when restricted to the domain $\{0,\dots,k\}$, f_1 is monotone decreasing to the left of i_0, constant between i_0 and i_1, and monotone increasing to the right of i_1. Note that $f_1(i) = |u| - l - r$, for $i_0 \le i \le i_1$.

On the other hand, observe that $f_2(i) = g(l-l_i) + g(r-r_i)$, where

$$g(x) := \frac{|x|+x}{2}, \text{ for } x \in \mathbb{Z};$$

satisfies $|g(x) - g(x-1)| \le 1$. In particular, if $w_{i+1} = \alpha$ then $|f_2(i+1) - f_2(i)| \le 1$ because $l_{i+1} = l_i + 1$ and $r_{i+1} = r_i$. Similarly, if $w_{i+1} = \beta$ then $|f_2(i+1) - f_2(i)| \le 1$ because $l_{i+1} = l_i$ and $r_{i+1} = r_i - 1$. Finally, if $w_{i+1} \notin \{\alpha, \beta\}$ then $l_{i+1} = l_i$ and $r_{i+1} = r_i$, hence $f_2(i+1) = f_2(i)$. In either case, we find that $|f_2(i+1) - f_2(i)| \le 1$ for $0 \le i < k$. As a result, since f_1 is integer-valued, $f_1 + f_2$ is decreasing for $i \le i_0$ but increasing for $i_1 \le i$, from which the lemma follows. □

Efficient algorithmic calculation. The proof of Lemma 3.2 can be adapted into a method (see Algorithm 1) that finds the distance between an arbitrary string w to a string of the form $v = \alpha^l\beta^r$ in $O(|w|)$ time—assuming that α, β, l, and r are known in advance. The algorithm exploits that $f_1(i)$ is constant for $i_0 \le i \le i_1$, reducing the calculation of $\ell(w, v)$ to minimizing f_2 over the restricted domain. This can be done through a loop where $f_2(i_0)$ can be found directly, and the remaining values can be found recursively by finding $f_2(i+1) - f_2(i)$ through cases depending on l_i, r_i, and w_{i+1}. This is faster than standard methods of finding the edit distance between strings with $O(|w||v|)$ time complexity.

A number of papers suggest methods for effectively computing the edit distance between run-length encoded strings [1, 18]. These methods adapt the standard dynamic programming approach to compute $\ell(u, v)$ in $O(r(u)|v| + r(v)|u|)$ time. Comparatively, Algorithm 1 has a few benefits and quirks: it assumes only one string is run-length encoded, it is fast due to specificity, and it provides a formula that is useful for proofs.

Algorithm 1 For computing the edit distance to a two-run string

Input. w a string, $\alpha \ne \beta$ alphabet characters, and $l, r > 0$ integers
Output. $\ell(w, \alpha^l\beta^r)$
$k \leftarrow |w|$
$i_0 \leftarrow \max\{0, \min\{l, k-r\}\}$
$i_1 \leftarrow \min\{k, \max\{l, k-r\}\}$
$l_i \leftarrow N_\alpha(w_1 \cdots w_{i_0})$
$r_i \leftarrow N_\beta(w_{i_0+1} \cdots w_k)$
$f_2 \leftarrow (|l-l_i| + l - l_i)/2 + (|r-r_i| + r - r_i)/2$
$m \leftarrow f_2$
for $i = i_0 + 1$ to i_1 **do**
  if $w_i = \beta$ **then**
    if $r_i \le r$ **then**
      $f_2 \leftarrow f_2 + 1$
    end if
    $r_i \leftarrow r_i - 1$
  end if
  if $w_i = \alpha$ and $l_i < l$ **then**
    $f_2 \leftarrow f_2 - 1$
    $m \leftarrow \min\{m, f_2\}$
    $l_i \leftarrow l_i + 1$
  end if
end for
$f_1 \leftarrow (k - l - r)/2 + (|k - l - r|)/2$
return $f_1 + m$

4. Metric dimension of Levenshtein graphs

Recall that a subset of nodes R in a graph G is said to resolve it when R resolves all pairs of different nodes, namely, for all nodes u and v, with $u \ne v$, there exists $r \in R$ such that $d(u, r) \ne d(v, r)$. The metric dimension of the graph, $\beta(G)$, is the size of its smallest resolving set.

The main result in this section is the following bound on the metric dimension of Levenshtein graphs.

Theorem 4.1. $O\Big(\frac{k_2}{\log_a k_2}\Big) \le \beta(\mathbb{L}_{k_1,k_2;a}) \le O\big(a\big((k_2+1)^2 - k_1^2\big)\big)$. *In particular, if* $\Delta := k_2 - k_1 + 1$ *then* $\beta(\mathbb{L}_{k_1,k_2;a}) = O(ak_2\Delta)$.

Observe that if $\Delta = \Theta(k_2)$ then $\beta(\mathbb{L}_{k_1,k_2;a})$ grows at most quadratically in terms of the maximum string length k_2. However, if $\Delta = \Theta(1)$ then $\beta(\mathbb{L}_{k_1,k_2;a})$ grows linearly with the largest string length. By setting $k_1 = k_2$, Theorem 4.1 may be applied to Hamming graphs as well. In this case, the lower bound of the theorem is tight because $\beta(\mathbb{H}_{k,a}) \sim 2k/\log_a(k)$ [13].

The remainder of this section is devoted to proving Theorem 4.1. The lower-bound is almost immediate from [14, Theorem 3.6]; nevertheless, for the sake of a self-contained exposition, we include its proof here. Indeed, if $R = \{r_1, \dots r_\beta\}$ is a resolving set of cardinality $\beta := \beta(\mathbb{L}_{k_1,k_2;a})$, then the transformation $d(v|R) := \big(\ell(v,r_1), \dots, \ell(v,r_\beta)\big)$ is one-to-one. Hence, since $0 \le \ell(v,r) \le \max\{|v|,|r|\} \le k_2$, for all $v, r \in V_{k_1,k_2;a}$, we must have that

$$a^{k_2} \le |V_{k_1,k_2;a}| \le (k_2+1)^\beta,$$

from which the left-hand side inequality in Theorem 4.1 follows. (In the above argument the inequality $|V_{k_1,k_2;a}| \ge a^{k_2}$, which neglects the parameter k_1, may seem absurdly loose; however, this is not the case because $|V_{k_1,k_2;a}| \le 2a^{k_2}$.)

The upper-bound in Theorem 4.1 follows directly from the following three results.

Lemma 4.1. Let $k_1 \le k \le k_2$. In $\mathbb{L}_{k_1,k_2;a}$, the following subset of nodes resolves any pair of different strings of length k:

$$R_{k,a} := \bigcup_{n=0}^{\lfloor a/2 \rfloor - 1} \Big\{ (2n)^i (2n+1)^{k-i} : 0 \le i \le k \Big\}. \tag{8}$$

Proof. Let $u = u_1 \cdots u_k$ and $v = v_1 \cdots v_k$ be nodes in $\mathbb{L}_{k_1,k_2;a}$ of the same length k that differ at certain position j. Define $\alpha := u_j$. Without loss of generality assume that $\alpha \ne (a-1)$ when a is odd.

Due to Theorem 2.1, the distance between pairs of nodes in $\mathbb{L}_{k_1,k_2;a}$ is either their Hamming or edit distance. But, since nodes in $R_{k,a}$ have at most two runs, Corollary 3.1 implies that $\ell(u,r) = h(u,r)$ and $\ell(v,r) = h(v,r)$, for each $r \in R_{k,a}$. Hence, the distance between u and v to any node in $R_{k,a}$ is always the Hamming distance.

If α is even, we claim that $\{\alpha^{j-1}(\alpha+1)^{k-j+1}, \alpha^j(\alpha+1)^{k-j}\}$ resolves u and v. By contradiction suppose otherwise, i.e. assume that $d(u, \alpha^{j-1}(\alpha+1)^{k-j+1}) = d(v, \alpha^{j-1}(\alpha+1)^{k-j+1})$ and $d(u, \alpha^j(\alpha+1)^{k-j}) = d(v, \alpha^j(\alpha+1)^{k-j})$. Define $\delta := d(u, \alpha^{j-1}(\alpha+1)^{k-j+1}) = d(v, \alpha^{j-1}(\alpha+1)^{k-j+1})$. Then

$$\begin{aligned} d(u, \alpha^j(\alpha+1)^{k-j}) &= h(u, \alpha^j(\alpha+1)^{k-j}) \\ &= \sum_{i=1}^{j-1} [\![u_i \ne \alpha]\!] + [\![u_j \ne \alpha]\!] \; + \sum_{i=j+1}^{k} [\![u_i \ne \alpha+1]\!] \pm [\![u_j \ne \alpha+1]\!] \\ &= h(u, \alpha^{j-1}(\alpha+1)^{k-j+1}) - 1 \\ &= \delta - 1. \end{aligned}$$

On the other hand, since $v_j \ne \alpha$:

$$\begin{aligned} d(v, \alpha^j(\alpha+1)^{k-j}) &= h(v, \alpha^j(\alpha+1)^{k-j}) \\ &= \sum_{i=1}^{j-1} [\![v_i \ne \alpha]\!] + [\![v_j \ne \alpha]\!] \; + \sum_{i=j+1}^{k} [\![v_i \ne \alpha+1]\!] \pm [\![v_j \ne \alpha+1]\!] \\ &= h(v, \alpha^{j-1}(\alpha+1)^{k-j+1}) + 1 - [\![v_j \ne \alpha+1]\!] \\ &\ge \delta, \end{aligned}$$

implying that $d(u, \alpha^j(\alpha+1)^{k-j}) \ne d(v, \alpha^j(\alpha+1)^{k-j})$, which is not possible. So, $\{\alpha^{j-1}(\alpha+1)^{k-j+1}, \alpha^j(\alpha+1)^{k-j}\}$ resolves u and v.

Likewise, if α is odd, one can show that $\{(\alpha-1)^{i-1}\alpha^{k-i+1}, (\alpha-1)^i\alpha^{k-i}\}$ resolves u and v, from which the lemma follows. □

Lemma 4.2. If θ is the string bijection induced by the transformation $\theta(\alpha) := (\alpha+1) \pmod a$, for $\alpha \in \{0, \dots, a-1\}$, then the set $\theta(R_{k-1;a}) \cup R_{k+1;a}$ resolves all pairs of different strings of length k that are permutations of each other.

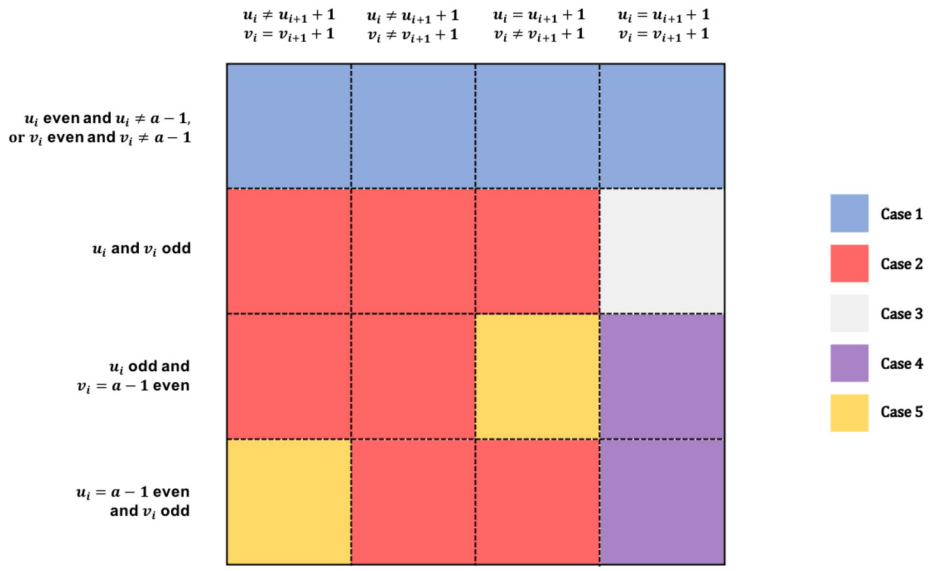


Fig. 3. Diagram associated with the different cases in the proof of Lemma 4.2.

Proof. Recall that $w_{(n)}$ and $w^{(n)}$ denote the prefix and suffix of a string w of length n, respectively.

Let u be a string of length $k > 1$, and $v \neq u$ correspond to a permutation of the characters in u. Let i be the first position at which u and v differ; in particular, $u_{(i-1)} = v_{(i-1)}$, and $u^{(k-i+1)}$ and $v^{(k-i+1)}$ are permutations of each other. We show the lemma by cases, see Fig. 3.

Case 1: Without loss of generality assume that u_i even and $u_i \neq (a-1)$. Define $\alpha := u_i$; in particular, $\alpha^i(\alpha+1)^{k+1-i} \in R_{k+1;a}$. We claim that the later string resolves u and v. Indeed, we may define

$$\lambda := N_\alpha\big(u_{(i-1)}\big) = N_\alpha\big(v_{(i-1)}\big);$$
$$\gamma := N_{\alpha+1}\big(u^{(k-i+1)}\big) = N_{\alpha+1}\big(v^{(k-i+1)}\big).$$

Next, using Lemmas 3.2 and 3.1 we find that

$$\begin{aligned}
&\ell(u, \alpha^i(\alpha+1)^{k+1-i})\\
&= \min\{\ell(u_{(i-1)}, \alpha^i) + \ell(u^{(k-i+1)}, (\alpha+1)^{k+1-i}), \ell(u_{(i)}, \alpha^i) + \ell(u^{(k-i)}, (\alpha+1)^{k+1-i})\}\\
&\le \ell(u_{(i)}, \alpha^i) + \ell(u^{(k-i)}, (\alpha+1)^{k+1-i})\\
&= k - \lambda - \gamma,
\end{aligned}$$

where for the second identity we have used that $u_i = \alpha$. Similarly, using that $v_i \neq \alpha$ we obtain that

$$\ell(v, \alpha^i(\alpha+1)^{(k+1)-i}) = \min\{k+1-\lambda-\gamma, k+1-\lambda-\gamma+[\![v_i = \alpha+1]\!]\} = k+1-\lambda-\gamma,$$

which shows the lemma for the Case 1.

We emphasize that Case 1 is the only one required for $a = 2$. In particular, without any loss of generality we may assume in what remains of this proof that $a \ge 3$.

Case 2: Without loss of generality assume that $u_i \neq u_{i+1}+1$ and that u_i and v_i are odd, or that u_i is odd and $v_i = a-1$ is even. Define $\alpha := u_i - 1$; in particular, $u_{i+1} \neq \alpha$ and $\alpha^{i+1}(\alpha+1)^{k-i} \in R_{k+1;a}$. We claim u and v are resolved by the later string. Indeed, preserving the definitions of λ and γ from Case 1, and using similar arguments to the ones used for that case, we find now that

$$\ell(u, \alpha^{i+1}(\alpha+1)^{k-i}) = \min\{k+2-\lambda-\gamma, k+2-\lambda-\gamma+[\![u_{i+1} = \alpha+1]\!]\} = k+2-\lambda-\gamma.$$

On the other hand, note that $v_i \neq \alpha$ otherwise $u_i = a$, which is not possible. Hence, using that $v_i \neq \alpha$ we obtain that

$$\ell(v, \alpha^{i+1}(\alpha+1)^{k-i}) = \min\{k+1-\lambda-\gamma, \ell(v_{(i+1)}, \alpha^i) + \ell(v^{(k-i-1)}, \beta^{k-i-1})\} \le k+1-\lambda-\gamma,$$

which shows the lemma for the Case 2.

Case 3: u_i and v_i odd, $u_i = u_{i+1}+1$, and $v_i = v_{i+1}+1$. Define $\alpha := u_i$ and $\beta := \theta(\alpha)$. We claim that $\alpha^i\beta^{k-i-1} \in \theta(R_{k-1;a})$ resolves u and v. To show so define

$$\lambda' := N_\alpha(u_{(i-1)}) = N_\alpha(v_{(i-1)});$$
$$\gamma' := N_\beta(u^{(k-i+1)}) = N_\beta(v^{(k-i+1)}).$$

Note that $u_{i+1} \neq \alpha$ and $u_{i+1} \neq \beta$ because $a \geq 3$; in particular, $N_\alpha(u_{(i+1)}) \leq i$ and $N_\beta(u^{(k-i)}) \leq k-i-1$. As a result, due to Lemmas 3.2-3.1, we find that

$$\ell(u, \alpha^i\beta^{k-i-1}) \leq \ell(u_{(i)}, \alpha^i) + \ell(u^{(k-i)}, \beta^{k-i-1}) = k - \lambda' - \gamma' - 1.$$

Likewise:

$$\ell(v, \alpha^i\beta^{k-i-1}) = \min\{\ell(v_{(i)}, \alpha^i) + \ell(v^{(k-i)}, \beta^{k-i-1}), \ell(v_{(i+1)}, \alpha^i) + \ell(v^{(k-i-1)}, \beta^{k-i-1})\}.$$

But note that $N_\alpha(v_{(i+1)}) \leq i$ because α is odd and v_{i+1} even, and $N_\beta(v^{(k-i)}) = N_\beta(u^{(k-i)}) \leq k-i-1$ because $u^{(k-i+1)}$ and $v^{(k-i+1)}$ are permutations of each other and $u_i, v_i \neq \beta$. Finally, since $v_i \neq \alpha$ and $v_{i+1} \neq \alpha$, we obtain that

$$\ell(v, \alpha^i\beta^{k-i-1}) = \min\{k - \lambda' - \gamma', k - \lambda' - \gamma' + [\![v_{i+1} = \beta]\!]\} = k - \lambda' - \gamma,$$

which shows the lemma for the Case 3.

Case 4. Without loss of generality assume that $u_i = u_{i+1} + 1$ is odd and that $v_i = v_{i+1} + 1 = a - 1$ is even. In particular, a is odd and $\alpha^i\beta^{k-1-i} \in \theta(R_{k-1;a})$ where $\alpha := u_i$ and $\beta := \alpha + 1$. We claim that $\alpha^i\beta^{k-1-i}$ resolves u and v. To see this, note that $u_{i+1} \notin \{\alpha, \alpha+1\}$; specifically, $N_\alpha(u_{(i+1)}) \leq i$ and $N_\beta(u^{(k-i)}) \leq k-i-1$. So, if λ' and γ' are as in Case 3 then Lemma 3.2 and Lemma 3.1 imply that

$$\ell(u, \alpha^i\beta^{k-i-1}) \leq \ell(u_{(i)}, \alpha^i) + \ell(u^{(k-i)}, \beta^{k-i-1}) = k - \lambda' - \gamma' - 1.$$

On the other hand, $v_i \neq \alpha$ hence $N_\alpha(v_{(i+1)}) \leq i$. Additionally, there must be some $v_j = u_{i+1}$ for some $j > i$, so $N_\beta(u^{(k-i)}) \leq k-i-1$. Thus:

$$\begin{aligned}
&\ell(v, \alpha^i\beta^{k-i-1}) \\
&= \min\{\ell(v_{(i)}, \alpha^i) + \ell(v^{(k-i)}, \beta^{k-i-1}), \ell(v_{(i+1)}, \alpha^i) + \ell(v^{(k-i-1)}, \beta^{k-i-1})\} \\
&= \min\{k - \lambda' - \gamma' + [\![v_i = \beta]\!], k - \lambda' - \gamma' + [\![v_i = \beta]\!] - [\![v_{i+1} = \alpha]\!]\} \\
&= k - \lambda' - \gamma',
\end{aligned}$$

where for the final identity we have used that $[\![v_{i+1} = \alpha]\!] = [\![v_i = \beta]\!]$. This shows the lemma for the Case 4.

Case 5. Without loss of generality assume that $u_i = u_{i+1} + 1$ is odd and $v_i = a - 1 \neq v_{i+1} + 1$ is even. In particular, a is odd and $(a-2)^i(a-1)^{k-i-1} \in \theta(R_{k-1;a})$. We claim that $(a-2)^i(a-1)^{k-i-1}$ resolves u and v. To show so, define

$$\lambda'' := N_{a-2}(u_{(i-1)}) = N_{a-2}(v_{(i-1)});$$
$$\gamma'' := N_{a-1}(u^{(k-i+1)}) = N_{a-1}(v^{(k-i+1)}).$$

Observe that $0 \leq u_{i+1} < u_i \leq a-2$ so $u_{i+1} \neq a-1$. As a result, due to Lemma 3.2-3.1:

$$\begin{aligned}
\ell(u, (a-2)^i(a-1)^{k-i-1}) &\leq \ell(u_{(i)}, (a-2)^i) + \ell(u^{(k-i)}, (a-1)^{k-i-1}\} \\
&= k - \lambda'' - \gamma'' - [\![u_i = a-2]\!] \\
&\leq k - \lambda'' - \gamma''.
\end{aligned}$$

On the other hand, since $v_i = a-1$, $N_{a-2}(v_{(i+1)}) \leq i$. Additionally, $v_{i+1} \neq a-2$. So:

$$\begin{aligned}
&\ell(u, (a-2)^i(a-1)^{k-i-1}) \\
&= \min\{\ell(v_{(i)}, (a-2)^i) + \ell(v^{(k-i)}, (a-1)^{k-i-1}), \ell(v_{(i+1)}, (a-2)^i) + \ell(v^{(k-i-1)}, (a-1)^{k-i-1})\} \\
&= \min\{k - \lambda'' - \gamma'' + 1, k - \lambda'' - \gamma'' + 1 + [\![v_{i+1} = a-1]\!]\} \\
&= k - \lambda'' - \gamma'' + 1,
\end{aligned}$$

which completes the proof of the lemma. □

Corollary 4.1. *$\mathbb{L}_{k_1,k_2,a}$ is resolved by a set of size $O\left(a\left((k_2+1)^2 - k_1^2\right)\right)$.*

Proof. Let θ be the character bijection defined in Lemma 4.2. Consider the sets

$$R_0 := \{0^{k_2}, \ldots, (a-1)^{k_2}\};$$
$$R_1 := \bigcup_{i=0}^{\lfloor (k_2-k_1)/2 \rfloor} \theta^i(R_{k_2-2i;a}) \cup \begin{cases} \emptyset, & k_2-k_1 \text{ even;} \\ R_{k_1;a}, & k_2-k_1 \text{ odd.} \end{cases}$$

We claim that $R := R_0 \cup R_1$ resolves $\mathbb{L}_{k_1,k_2;a}$. For this, let u and v be different nodes in this Levenshtein graph. We show that R resolves these nodes by considering different cases.

First, suppose that u and v are not permutations of each other; in particular, for some alphabet character α, $N_\alpha(u) \neq N_\alpha(v)$. If $|u| = |v|$ then, due to Lemma 3.1, $\ell(u, \alpha^{k_2}) = k_2 - N_\alpha(u) \neq k_2 - N_\alpha(v) = \ell(v, \alpha^{k_2})$ i.e. u and v are resolved. Instead, if $|u| \neq |v|$ and R_0 did not resolve them, then

$$|u| = \sum_{\alpha=0}^{a-1} N_\alpha(u) = \sum_{\alpha=0}^{a-1} (k_2 - \ell(\alpha^{k_2}, u)) = \sum_{\alpha=0}^{a-1} (k_2 - \ell(\alpha^{k_2}, v)) = \sum_{\alpha=0}^{a-1} N_\alpha(v) = |v|,$$

which is not possible. Hence R_0 resolves all pairs of nodes in $\mathbb{L}_{k_1,k_2;a}$ that are not permutations of each other.

Next, suppose that $u \neq v$ are permutations of each other. Let $k := |u| = |v|$. If $k_2 - k$ is even or $k = k_1$ then $\theta^i(R_{k;a}) \subset R$ for some integer $0 \leq i \leq \lfloor (k_2-k_1)/2 \rfloor$. Further, since θ is an automorphism, $u_0 := \theta^{-i}(u)$ and $v_0 := \theta^{-i}(v)$ are distinct strings of the same length k, and the distances from u and v to the nodes in $\theta^i(R_{k;a})$ are the same as those from u_0 and v_0 to $R_{k;a}$. But, due to Lemma 4.1, u_0 and v_0 are resolved by $R_{k;a}$, so u and v are resolved by $\theta^i(R_{k;a})$.

Instead, if $k_2 - k$ is odd and $k \neq k_1$ then $\theta^{i+1}(R_{k-1;a}) \cup \theta^i(R_{k+1;a}) \subset R$ for some integer $0 \leq i < \lfloor (k_2-k_1)/2 \rfloor$. But $u_0 := \theta^{-i}(u)$ and $v_0 := \theta^{-i}(u)$ are also permutations of each other so, by Lemma 4.2, u_0 and v_0 are resolved by $\theta(R_{k-1;a}) \cup R_{k+1;a}$. Hence, since θ is an automorphism, u and v are resolved by $\theta^{i+1}(R_{k-1;a}) \cup \theta^i(R_{k+1;a})$. This shows that R resolves $\mathbb{L}_{k_1,k_2;a}$.

Finally, observe that

$$|R_{k,a}| = \begin{cases} 1, & \text{if } k = 0; \\ \lfloor \frac{a}{2} \rfloor (k+1), & \text{if } k > 0. \end{cases}$$

Therefore

$$\begin{aligned} |R| &\leq |R_0| + |R_{k_1;a}| + \sum_{i=0}^{\lfloor \frac{k_2-k_1}{2} \rfloor} |\theta^i(R_{k_2-2i;a})| \\ &= a + \left\lfloor \frac{a}{2} \right\rfloor (k_1+1) + \left\lfloor \frac{a}{2} \right\rfloor \sum_{i=0}^{\lfloor \frac{k_2-k_1}{2} \rfloor} (k_2 - 2i + 1) \\ &= O\big(a(k_2+1)(k_2-k_1+1)\big) \\ &= O\left(a\big((k_2+1)^2 - k_1^2\big)\right), \end{aligned}$$

from which the result follows. □

5. Automorphisms of Levenshtein graphs

In what follows, $\mathbb{A}(G)$ denotes the automorphism group of a graph G.

In addition, ρ denotes the string reversal, i.e. if $u = u_1 \cdots u_k$ is a string of length $k \geq 1$ then $\rho(u) := u_k \cdots u_1$. By definition, $\rho(\varepsilon) := \varepsilon$. On the other hand, given an alphabet bijection $\xi : \{0, \ldots, a-1\} \to \{0, \ldots, a-1\}$, we define $\xi(u) := \xi(u_1) \cdots \xi(u_k)$ and $\xi(\varepsilon) := \varepsilon$. We refer to any such transformation as a character bijection.

The main result in this section completes the characterization of automorphisms of Levenshtein graphs. The cases not covered by our result have implicitly been addressed in the literature. In fact, $\mathbb{L}_{0,1;a}$ is isomorphic to the complete graph K_{a+1}, whose automorphism group is the permutation group S_{a+1} (i.e. the set of all permutations of $\{0, \ldots, a\}$). In particular, $|\mathbb{A}(\mathbb{L}_{0,1;a})| = (a+1)!$. These Levenshtein graphs are somewhat degenerate in that they are the only Levenshtein graphs where automorphisms do not necessarily preserve string lengths.

On the other hand, $\mathbb{L}_{k,k;a}$ is isomorphic to the Hamming graph $\mathbb{H}_{k,a}$ (Lemma 2.3), whose automorphism group is $(\times_{i=1}^{k} S_a) \rtimes S_k$ [5,27]. In other words, the automorphisms of $\mathbb{L}_{k,k;a}$ are the composition of character permutations with character-wise alphabet bijections. Accordingly, $|\mathbb{A}(\mathbb{L}_{k,k;a})| = k! \cdot (a!)^k$.

The remaining Levenshtein graphs are addressed by our next result.

Theorem 5.1. *Let $k_1 \neq k_2$ and $k_2 \geq 2$. In $\mathbb{L}_{k_1,k_2;a}$, a node bijection σ is an automorphism if and only if σ is a character bijection, string reversal, or a composition of both. In particular, $\mathbb{L}_{k_1,k_2;a}$ has $a! \cdot 2$ automorphisms.*

The proof of this theorem is given at the end of this section. It is based on the following lemmas.

Lemma 5.1. The string reversal and character bijections are automorphisms of $\mathbb{L}_{k_1,k_2;a}$.

Proof. Let ξ be a character bijection. Since ξ and ρ preserve string lengths, $\xi(V_{k_1,k_2;a}) \subset V_{k_1,k_2;a}$ and $\rho(V_{k_1,k_2;a}) \subset V_{k_1,k_2;a}$. Furthermore, since the character bijection associated with the alphabet bijection ξ^{-1} is an inverse for ξ, and ρ is an involution, ξ and ρ are bijections from $V_{k_1,k_2;a}$ onto itself. It is convenient to extend ξ to strings formed from the enlarged alphabet $\{0,\ldots,a-1,\text{-}\}$, defining $\xi(\text{-}) = \text{-}$. Likewise, extend ρ to strings that may include gaps besides alphabet characters.

Let $u, v \in V_{k_1,k_2;a}$ and A an alignment of length $k \geq 1$ between them:

$$A = \begin{matrix} \alpha_1 & \ldots & \alpha_k \\ \beta_1 & \ldots & \beta_k \end{matrix}.$$

Define the following alignment between $\xi(u)$ and $\xi(v)$:

$$\xi(A) := \begin{matrix} \xi(\alpha_1) & \ldots & \xi(\alpha_k) \\ \xi(\beta_1) & \ldots & \xi(\beta_k) \end{matrix}.$$

Clearly, $\text{score}(\xi(A)) = \text{score}(A)$, which implies that $\ell(\xi(u),\xi(v)) \leq \ell(u,v)$, for all $u, v \in V_{k_1,k_2;a}$ and character bijection ξ. In particular, $\ell(\xi^{-1}(\xi(u)),\xi^{-1}(\xi(v))) \leq \ell(\xi(u),\xi(v))$, implying that $\ell(u,v) = \ell(\xi(u),\xi(v))$. A similar argument shows that $\ell(u,v) = \ell(\rho(u),\rho(v))$, which completes the proof. □

Next, we discuss the degree of nodes on the infinite graph $\mathbb{L}_a$. Our result can be generalized to arbitrary Levenshtein graphs by restricting the length of the neighbors of a node.

Recall that the number of runs in a node u is denoted $r(u)$. The next result may be regarded a corollary of the proof of [16, Theorem 1] in the context of binary strings and was stated without proof in [17]. We include its proof for the sake of completeness.

Lemma 5.2. ([16,17].) A node u on $\mathbb{L}_a$ has $r(u)$ neighbors of length $|u|-1$, $|u|(a-1)$ neighbors of length $|u|$, and $a+|u|(a-1)$ neighbors of length $|u|+1$. In particular, u has degree $a + r(u) + 2|u|(a-1)$.

Proof. Recall that substitutions keep the length of a node, whereas character deletions and insertions reduce and increase, respectively, its length by one unit. In particular, u has $|u|(a-1)$ neighbors of length $|u|$, and $r(u)$ neighbors of length $|u|-1$.

Let us now focus on the neighbors of u that can be reached due to a single insertion. An insertion may either keep or increase the number of runs. The former occurs only if a run is enlarged by one character, and there are $r(u)$ ways to do so. The latter occurs only if a run is split by a character into two, or two consecutive runs are separated by a single-character run, which can be done in $(|u|+1)(a-1)-(r(u)-1) = a+|u|(a-1)-r(u)$ ways. In particular, $r(u)+a+|u|(a-1)-r(u) = a+|u|(a-1)$ nodes can be reached from u through a single insertion. From this, the proposition follows. □

Lemma 5.3. If $k_1 + 1 < k_2$ then any automorphism of $\mathbb{L}_{k_1,k_2;a}$ preserves the length of strings of length k_2.

Proof. Let σ be an automorphism of $\mathbb{L}_{k_1,k_2;a}$ (recall the implicit assumption that $a \geq 2$). We claim that $\sigma(V_{k_2,k_2;a}) \subset V_{k_1,k_2-2;a} \cup V_{k_2,k_2;a}$. By contradiction suppose that there is a node u such $|u| = k_2$ and $|\sigma(u)| = k_2 - 1$. Then, due to Lemma 5.2:

$$\begin{aligned} \deg(u) &= r(u) + k_2(a-1); \\ \deg(\sigma(u)) &= r(\sigma(u)) + a + 2(k_2-1)(a-1). \end{aligned}$$

As a result, using that $1 \leq r(w) \leq |w|$ for any non-empty string w, we obtain that

$$\begin{aligned} deg(\sigma(u)) &\geq 1 + a + 2(k_2-1)(a-1) \\ &\geq 1 + a + (k_2-1)(a-1) + (k_2-1) \\ &= k_2 + k_2(a-1) + 1 \\ &> \deg(u), \end{aligned}$$

which is not possible because automorphisms preserve node degrees.

Finally, we show that $\sigma(V_{k_2,k_2;a}) = V_{k_2,k_2;a}$. For this note that no vertex in $V_{k_2,k_2-2;a}$ can be a neighbor of a vertex in $V_{k_2,k_2;a}$ because any alignment between a word of length k_2-2 and another of length k_2 must include at least two gaps. On the other hand, since $V_{k_2,k_2;a}$ is the vertex set of $\mathbb{H}_{k_2;a}$, which is a connected sub-graph of $\mathbb{L}_{k_1,k_2;a}$, $\sigma(V_{k_2,k_2;a})$ is the vertex set of a connected subgraph of $\mathbb{L}_{k_1,k_2;a}$. As a result, since $\sigma(V_{k_2,k_2;a}) \subset V_{k_1,k_2-2;a} \cup V_{k_2,k_2;a}$, either $\sigma(V_{k_2,k_2;a}) \subset V_{k_1,k_2-2;a}$ or $\sigma(V_{k_2,k_2;a}) \subset V_{k_1,k_2;a}$. Since the former inclusion is not possible because $|V_{k_1,k_2-2;a}| < |V_{k_1,k_2;a}|$, we must have $\sigma(V_{k_2,k_2;a}) \subset V_{k_2,k_2;a}$, which shows the proposition. □

Lemma 5.4. Let $k_1 \neq k_2$ and $k_2 \geq 2$, and define $X := \{0^{k_2}, \ldots, (a-1)^{k_2}\}$. If σ is an automorphism of $\mathbb{L}_{k_1,k_2;a}$ then $\sigma(X) = X$.

Proof. Let σ be an automorphism of $\mathbb{L}_{k_1,k_2;a}$.

We first show that $\sigma(X) \subset V_{k_2,k_2;a}$. Due to Lemma 5.3, this is direct when $k_1+1 < k_2$. Hence assume that $k_1+1 = k_2$; in particular, $V_{k_1,k_2;a} = V_{k_1,k_1;a} \cup V_{k_2,k_2;a}$. Suppose that $\sigma(X) \cap V_{k_1,k_1;a} \neq \emptyset$. Then, there would be $x \in X$ such that $|\sigma(x)| = k_1$. In particular, due to Lemma 5.2, it would follow that

$$\begin{aligned}\deg(\sigma(x)) &= a + 2(k_2-1)(a-1)\\ &> a + (k_2-1)(a-1)\\ &= 1 + k_2(a-1)\\ &= \deg(x),\end{aligned}$$

which it is not possible because automorphisms preserve node degrees. As a result, $\sigma(X) \cap V_{k_1,k_1;a} = \emptyset$, i.e. $\sigma(X) \subset V_{k_2,k_2;a}$, which shows the claim.

Finally, since $\sigma(X) \subset V_{k_2,k_2;a}$, for each $x \in X$, Lemma 5.2 implies that $\deg(x) = 1 + k_2(a-1)$ and $\deg(\sigma(x)) = r(\sigma(x)) + k_2(a-1)$. Since $\deg(x) = \deg(\sigma(x))$, we must have $r(\sigma(x)) = 1$, i.e. $\sigma(x) \in X$, which shows the lemma. □

Lemma 5.5. Let $k_1 \neq k_2$ and $k_2 \geq 2$. If σ is an automorphism of $\mathbb{L}_{k_1,k_2;a}$ then the following properties apply:

1. There is a character bijection ξ such that, for every alphabet character α and string $u \in V_{k_1,k_2;a}$, $N_\alpha(u) = N_{\xi(\alpha)}(\sigma(u))$; in particular, $\sigma(\alpha^k) = \xi(\alpha)^k$ for each alphabet character α and $k_1 \leq k \leq k_2$.
2. For all $u \in V_{k_1,k_2;a}$, $|\sigma(u)| = |u|$.
3. For all $u \in V_{k_1,k_2;a}$ with $|u| = k_2$, $r(\sigma(u)) = r(u)$.

Proof. Consider an automorphism σ of $\mathbb{L}_{k_1,k_2;a}$, and let X be as in Lemma 5.4. In particular, $\sigma(X) = X$. Since σ is bijective, there exists an alphabet bijection $\xi : \{0,\ldots,a-1\} \to \{0,\ldots,a-1\}$ such that $\sigma(x) = \xi(x)^{k_2}$, for each $x \in X$. As before, we denote the automorphism associated with ξ with the same symbol.

Let α be an alphabet character, and u a node in $\mathbb{L}_{k_1,k_2;a}$. Since $\alpha^{k_2} \in X$, it follows from Lemma 3.1 that

$$\ell(\sigma(u), \sigma(\alpha^{k_2})) = \ell(\sigma(u), \xi(\alpha)^{k_2}) = k_2 - N_{\xi(\alpha)}(\sigma(u)).$$

Since $\ell(u, \alpha^{k_2}) = k_2 - N_\alpha(u)$, and we must have $\ell(u,\alpha^{k_2}) = \ell(\sigma(u), \sigma(\alpha^{k_2}))$, Property 1 follows. From this, Property 2 is immediate because

$$|u| = \sum_{\alpha=0}^{a-1} N_\alpha(u) = \sum_{\alpha=0}^{a-1} N_{\xi(\alpha)}(\sigma(u)) = |\sigma(u)|.$$

Finally, due to Property 2 and Lemma 3.1, if $|u| = k_2$ then $\deg(\sigma(u)) = r(\sigma(u)) + k_2(a-1)$. Likewise, $\deg(u) = r(u) + k_2(a-1)$. In particular, $r(u) = r(\sigma(u))$ because $\deg(u) = \deg(\sigma(u))$, which shows Property 3. □

Proof of Theorem 5.1. Let σ be an automorphism of $\mathbb{L}_{k_1,k_2;a}$, and ξ be the corresponding character bijection described in Lemma 5.5. Observe that $(\xi^{-1} \circ \sigma)$ preserves character counts because, due to property (1) in the lemma, $N_\alpha(u) = N_\alpha((\xi^{-1} \circ \sigma)(u))$ for each character α and $u \in V_{k_1,k_2;a}$.

Next observe the string $0^{k_2-1}1$. From properties (2) and (3) in Lemma 5.5, we find that $(\xi^{-1} \circ \sigma)(0^{k_2-1}1)$ is a string of length k_2 with two runs. In particular, since $(\xi^{-1} \circ \sigma)$ preserves character counts, $(\xi^{-1} \circ \sigma)(0^{k_2-1}1) \in \{0^{k_2-1}1, 10^{k_2-1}\}$. If $(\xi^{-1} \circ \sigma)(0^{k_2-1}1) = 10^{k_2-1}$, define $\psi := \rho$, otherwise define ψ to be the identity. In either case, ψ is its own inverse; in particular, if we define

$$\iota := \psi \circ \xi^{-1} \circ \sigma = \psi^{-1} \circ \xi^{-1} \circ \sigma,$$

then

$$\iota(0^{k_2-1}1) = 0^{k_2-1}1. \tag{9}$$

We aim to show next that ι is the identity, focusing first on strings of length k_2 with two runs. In fact, note that ι preserves character and run counts because ψ and $(\xi^{-1}\circ\sigma)$ do. Hence, if $\alpha\neq\beta$ are characters and $0<k<k_2$ then

$$\iota(\alpha^{k_2-k}\beta^k)\in\{\alpha^{k_2-k}\beta^k,\,\beta^k\alpha^{k_2-k}\}. \tag{10}$$

First, let $\alpha=0$ and $\beta=1$. Assume that $\iota(0^{k_2-k}1^k)=1^k0^{k_2-k}$ for some $0<k<k_2$. Then, using Theorem 2.1, Corollary 3.1, and Equation (9), we find the following distances are

$$\begin{aligned} d(0^{k_2-k}1^k,0^{k_2-1}1)&=h(0^{k_2-k}1^k,0^{k_2-1}1)=k-1;\\ d(\iota(0^{k_2-k}1^k),\iota(0^{k_2-1}1))&=h(1^k0^{k_2-k},0^{k_2-1}1)=k+1;\end{aligned}$$

which is not possible because automorphisms preserve distances. Thus $\iota(0^{k_2-k}1^k)=0^{k_2-k}1^k$, for all $0<k<k_2$.

Second, if $\alpha=1$, $\beta=0$, and $\iota(1^{k_2-k}0^k)=0^k1^{k_2-k}$ for some $0<k<k_2$, then $\iota(1^{k_2-k}0^k)=0^k1^{k_2-k}=\iota(0^k1^{k_2-k})$, which is not possible because ι is one-to-one. Therefore $\iota(1^{k_2-k}0^k)=1^{k_2-k}0^k$, for all $0<k<k_2$.

Third, let $\alpha\neq1$ and $\beta=1$. Assume that $\iota(\alpha^{k_2-k}1^k)\neq\alpha^{k_2-k}1^k$ for some $0<k<k_2$. Then, due to Equation (10):

$$d(\alpha^{k_2-k}1^k,0^{k_2-k}1^k)=h(\alpha^{k_2-k}1^k,0^{k_2-k}1^k)=(k_2-k)[\![\alpha\neq0]\!];$$

$$d(\iota(\alpha^{k_2-k}1^k),\iota(0^{k_2-k}1^k))=h(1^k\alpha^{k_2-k},0^{k_2-k}1^k)=\begin{cases}k_2, & 0<k<k_2/2 \text{ and } \alpha\neq0;\\ 2k, & 0<k<k_2/2 \text{ and } \alpha=0;\\ 2(k_2-k), & k_2/2\leq k<k_2.\end{cases}$$

In particular, $d(\alpha^{k_2-k}1^k,0^{k_2-k}1^k)\neq d(\iota(\alpha^{k_2-k}1^k),\iota(0^{k_2-k}1^k))$, which is a contradiction because ι must preserve distances. So, $\iota(\alpha^{k_2-k}1^k)=\alpha^{k_2-k}1^k$ for all $\alpha\neq1$ and $0<k<k_2$.

Finally, let $\alpha\neq\beta$ be arbitrary characters in the alphabet. If $\alpha=1$ let $\gamma=0$, otherwise let $\gamma=1$. Through our second and third cases we have shown that $\iota(\alpha^{k_2-k}\gamma^k)=\alpha^{k_2-k}\gamma^k$ for all $0<k<k_2$. Next, assume that $\iota(\alpha^{k_2-k}\beta^k)\neq\alpha^{k_2-k}\beta^k$ for some $0<k<k_2$. Then, as we have argued before we find that:

$$d(\alpha^{k_2-k}\beta^k,\alpha^{k_2-k}\gamma^k)=h(\alpha^{k_2-k}\beta^k,\alpha^{k_2-k}\gamma^k)=k[\![\beta\neq\gamma]\!];$$

$$d(\iota(\alpha^{k_2-k}\beta^k),\iota(\alpha^{k_2-k}\gamma^k))=h(\beta^k\alpha^{k_2-k},\alpha^{k_2-k}\gamma^k)=\begin{cases}k_2, & k_2/2\leq k<k_2 \text{ and } \beta\neq\gamma;\\ 2(k_2-k), & k_2/2\leq k<k_2 \text{ and } \beta=\gamma;\\ 2k, & 0<k<k_2/2.\end{cases}$$

But then, once again we find that $d(\alpha^{k_2-k}\beta^k,\alpha^{k_2-k}\gamma^k)\neq d(\iota(\alpha^{k_2-k}\beta^k),\iota(\alpha^{k_2-k}\gamma^k))$, which is not possible. Consequently, for all $\alpha\neq\beta$ and $0<k<k_2$, $\iota(\alpha^{k_2-k}\beta^k)=\alpha^{k_2-k}\beta^k$.

Thus far, we have shown that if u is a string where $|u|=k_2$ and $r(u)\leq2$ then $\iota(u)=u$.

Let $R_{k_2,a}=\{r_1,\ldots,r_n\}$ be as defined by Equation (8). Note, for any $r_i\in R_{k_2,a}$ that $|r_i|=k_2$ and $r(r_i)=k_2$, implying that $\iota(r_i)=r_i$. Further, from Lemma 4.1, the transformation $\Phi(u):=\big(d(u,r_1),\ldots,d(u,r_n)\big)$ is one-to-one over nodes of length k_2. Consider an arbitrary node u such that $|u|=k_2$. From Theorem 5.5, we know that $|\iota(u)|=k_2$. As a result:

$$\begin{aligned}\Phi(u)&=\big(d(u,r_1),\ldots,d(u,r_n)\big)\\ &=\big(d(\iota(u),\iota(r_1)),\ldots,d(\iota(u),\iota(r_n))\big)\\ &=\big(d(\iota(u),r_1),\ldots,d(\iota(u),r_n)\big)\\ &=\Phi\big(\iota(u)\big).\end{aligned}$$

In particular, since Φ is one-to-one over vectors of length k_2, $\iota(u)=u$ for all node u such that $|u|=k_2$.

Finally, we prove by induction in k, with $k_1\leq k\leq k_2$, that $\iota(v)=v$ for all $v\in V_{k,k_2;a}$. The base case with $k=k_2$ was just shown above. Next, consider a $k_1\leq k<k_2$ and suppose that $\iota(v)=v$, for all $v\in V_{k+1,k_2;a}$. If $k=0$, property 2 of Lemma 5.5 implies that $\iota(\epsilon)=\epsilon$; in particular, $\iota(v)=v$ for all $v\in V_{k,k_2;a}$. Instead, if $k>0$, consider a string u of length k. From Lemma 5.2, u has $a+|u|(a-1)\geq3$ neighbors of length $k+1$. Let v_1, v_2, and v_3 be different neighbors of u of length $k+1$. By the inductive hypothesis: $\iota(v_i)=v_i$, for $1\leq i\leq3$. So, since ι is an automorphism, v_1, v_2, and v_3 are also neighbors of $\iota(u)$. The end of the proof relies on the following result.

Lemma 5.6. (Adjusted from [17, Theorem 4].) A node v in $\mathbb{L}_a$ is uniquely determined by three of its different neighbors of length $|v|+1$.

The lemma implies that $\iota(u)=u$ for all $|u|=k$, i.e. $\iota(v)=v$ for all $v\in V_{k,k_2;a}$.

The above shows that $\iota=\psi^{-1}\circ\xi^{-1}\circ\sigma$ is the identity. In particular, $\sigma=\xi\circ\psi$, where ξ is a character bijection and ψ is either the string reversion or the identity, which completes the proof of Theorem 5.1. □

6. Determining number of Levenshtein graphs

For a graph $G = (V, E)$, a set of nodes $D \subset V$ is called determining when the identity is the only $\sigma \in \mathbb{A}(G)$ such that $\sigma(x) = x$, for all $x \in D$ (this is equivalent to the definition given at the end of the Introduction). The determining number of G, denoted $\mathrm{Det}(G)$, is the size of its smallest determining set. (A graph with a trivial automorphism group has a determining number of 0.)

We implicitly encountered determining sets of Levenshtein graphs in the proof of Theorem 5.1, which essentially uses that $\{0^{k_2}, \ldots, (a-1)^{k_2}, w\}$, with w any non-palindromic string such that $k_1 \leq |w| \leq k_2$, is a determining set of $\mathbb{L}_{k_1,k_2;a}$ when $k_1 \neq k_2$ and $k_2 \geq 2$.

Since $\mathbb{L}_{0,1;a}$ is isomorphic to K_{a+1}, it follows from [3] that $\mathrm{Det}(\mathbb{L}_{0,1;a}) = a$. On the other hand, since $\mathbb{L}_{k,k;a}$ is isomorphic to $\mathbb{H}_{k,a}$, which may be described as the Cartesian product of k copies of K_a, tight bounds on $\mathrm{Det}(\mathbb{L}_{k,k;a})$ follow from [4].

On the other hand, it can be shown by an exhaustive test that if $k_1 \neq k_2$ and $(k_2, a) = (2, 2)$ then $\mathrm{Det}(\mathbb{L}_{k_1,2;2}) = 2 > \lfloor a/k_2 \rfloor$. In this case, $\{01, 00\}$ is one of a few minimal determining sets. Our following result addresses the determining number of the remaining Levenshtein graphs.

Theorem 6.1. *If $k_1 \neq k_2$, $k_2 \geq 2$, and $(k_2, a) \neq (2, 2)$ then* $\mathrm{Det}(\mathbb{L}_{k_1,k_2;a}) = \left\lceil \frac{a}{k_2} \right\rceil$.

The remainder of this section is devoted to stating and proving two auxiliary results and showing this theorem.

Lemma 6.1. If $k_1 \neq k_2$ and $k_2 \geq 2$ then at least $(a-1)$ of the a alphabet characters must be represented in a determining set of $\mathbb{L}_{k_1,k_2;a}$.

Proof. Let $D = \{d_1, \ldots, d_n\}$, with $n \geq 1$, be a determining set, and S the set of alphabet characters that occur at least once in D, i.e., $S = \{(d_i)_j : 1 \leq i \leq n, 1 \leq j \leq |d_i|\}$. If $|S| < a-1$ then there would exist at least two distinct alphabet characters $\alpha, \beta \notin S$. Let μ be the character bijection that swaps α and β, i.e. $\mu(\alpha) = \beta$ and $\mu(\beta) = \alpha$, but acts as the identity on every other character. Then, $\mu(d) = d$, for all $d \in D$; in particular, since μ is not the identity, D could not be a determining set. Since this is not possible, $|S| \geq a-1$, which shows the lemma. □

Lemma 6.2. If $k_1 \neq k_2$ and $k_2 \geq 2$ then $\mathrm{Det}(\mathbb{L}_{k_1,k_2;a}) \geq \left\lceil \frac{a}{k_2} \right\rceil$.

Proof. Let $D = \{d_1, \ldots, d_n\}$, with $n \geq 1$, be a determining set, and S the set of alphabet characters that occur at least once in D. Define $\ell_0 = 0$ and $\ell_i = \sum_{j=1}^{i} |d_i|$ for $1 \leq i \leq n$.

We claim that $\ell_n \geq a$. By contradiction, assume that $\ell_n < a$. Since $\ell_n \geq |S|$, Lemma 6.1 implies that $\ell_n = |S| = a-1$. In particular, up to a character bijection, we may assume that $S = \{0, \ldots, a-2\}$, and that $d_i = \ell_{i-1} \ldots (\ell_i - 1)$ for $1 \leq i \leq n$. Consider the character bijection μ such that $\mu(a-1) = a-1$, and $\mu(j) = \ell_i + \ell_{i-1} - 1 - j$ for $\ell_{i-1} \leq j \leq \ell_i - 1$ and $1 \leq i \leq n$. In particular, μ acts as a reversal on each string in D. Then $(\mu \circ \rho)(d_i) = d_i$, for all $1 \leq i \leq n$, hence $(\mu \circ \rho)$ must be the identity. However, this is not possible because $(\mu \circ \rho)(0(a-1)) = (a-1)(a-2)$. Hence $\ell_n \geq a$, which implies the lemma because $n \cdot k_2 \geq \sum_{i=1}^{n} |d_i| = \ell_n \geq a$. □

Proof of Theorem 6.1. Define $n := \lceil \frac{a}{k_2} \rceil$; in particular, $n \geq 1$. Due to Lemma 6.2, it suffices to construct a determining set of size n, for which we consider three cases. First, if $k_2 \geq a$, define $D := \{d\}$ where

$$d := \begin{cases} 0^{k_2-1}1, & a = 2; \\ 0^{k_2-a+2}1 \cdots (a-2), & a \geq 3. \end{cases}$$

Since at least $a-1$ alphabet characters are represented in d, the identity is the only character bijection that preserves d. On the other hand, if $\sigma = \mu \circ \rho$, where μ is any character bijection then, for $a = 2$, $\sigma(d) = \mu(1)\mu(0)^{k_2-1}$ with $k_2 - 1 \geq 2$; in particular $\sigma(d) \neq d$. Similarly, if $a \geq 3$ then $\sigma(d) = \mu(a-2) \cdots \mu(1)\mu(0)^{k_2-a+2}$ with $k_2 - a + 2 \geq 2$, and again $\sigma(d) \neq d$. Therefore, D is a determining set.

Second, if $2 < k_2 < a$, let $D := \{d_1, \ldots, d_n\}$ be of cardinality n such that $d_1 := 0012 \ldots (k_2 - 2)$, $d_1, \ldots, d_n$ are of length k_2, and every character in $\{0, \ldots, a-2\}$ is used by at least one node in D. Since $a-1$ alphabet characters are represented in D, the identity is the only character bijection that maps each d_i to itself. However, if $\sigma = \mu \circ \rho$, where μ is any character bijection, then $\sigma(d_1) = \mu(k_2 - 2) \cdots \mu(1)\mu(0)^2 \neq d_1$. So, D is a determining set.

Finally, if $k_2 = 2$; in particular, $a \geq 3$, let $D = \{d_1, \ldots, d_n\}$ be of cardinality n such that $d_1 := 01$, $d_2 := 12$, $d_1, \ldots, d_n$ are of length 2, and every character in $\{0, \ldots, a-2\}$ is used by at least one node in D. Once again, since at least $a-1$ alphabet characters are represented in D, the identity is the only character bijection that maps each d_i to itself. Next, let $\sigma = \mu \circ \rho$, where μ is any character bijection. If $\sigma(01) = 01$ then $\mu(1) = 0$. If this is the case then $\sigma(12) = \mu(2)0 \neq 12$, i.e. either $\sigma(01) \neq 01$ or $\sigma(12) \neq 12$. Hence D is determining and the theorem follows. □

7. Conclusion

In this manuscript, we have introduced the notion of Levenshtein graphs, which generalize Hamming graphs but allow for nodes (words) of different lengths. The underlying motivation for this is to use resolving sets in Levenshtein graphs to represent words of varying size as points in Euclidean spaces using graph embeddings of the form given in Equation (1).

We have shown that Levenshtein graphs are connected; however, unlike Hamming graphs, their distance is not necessarily equal to edit distance between nodes. We have also bounded the metric dimension of Levenshtein graphs and constructed resolving sets composed only of two-run strings. This construction is based on novel formulas to compute the edit distance to any one-run or two-run string. In addition, we have thoroughly characterized their automorphism group and determining number.

It remains to characterize the metric dimension of Levenshtein graphs more explicitly, or at least asymptotically. A technical difficulty for this is the lack of symmetries of these graphs, exemplified by their relatively small automorphism group. Nevertheless, numerical trials through the ICH algorithm suggest that the actual metric dimension of $\mathbb{L}_{k;a}$ grows at most linearly with the maximal string length k. However, these trials are limited due to the exponential growth of Levenshtein graphs, and further theoretical findings may be necessary to settle this claim.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank the reviewers for their thorough reading and constructive remarks about our manuscript. This research was partially funded by NSF IIS grant 1836914.

References

[1] O. Arbell, G.M. Landau, J.S. Mitchell, Edit distance of run-length encoded strings, Inf. Process. Lett. 83 (6) (2002) 307–314.
[2] D. Bar-Lev, T. Etzion, E. Yaakobi, On Levenshtein balls with radius one, in: 2021 IEEE International Symposium on Information Theory (ISIT), IEEE, 2021, pp. 1979–1984.
[3] D.L. Boutin, Identifying graph automorphisms using determining sets, Electron. J. Comb. (2006) R78.
[4] D.L. Boutin, The determining number of a Cartesian product, J. Graph Theory 61 (2) (2009) 77–87.
[5] F.A. Chaouche, A. Berrachedi, Automorphisms group of generalized Hamming graphs, in: Fifth Cracow Conference on Graph Theory USTRON '06, in: Electronic Notes in Discrete Mathematics, vol. 24, 2006, pp. 9–15.
[6] S.A. Cook, The complexity of theorem-proving procedures, in: Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71, ACM, New York, NY, USA, 1971, pp. 151–158.
[7] R. Durbin, S.R. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, 1998.
[8] D. Erwin, F. Harary, Destroying automorphisms by fixing nodes, Discrete Math. 306 (24) (2006) 3244–3252.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., New York, NY, USA, 1979.
[10] A. Grover, J. Leskovec, Node2vec: scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864.
[11] F. Harary, R.A. Melter, On the metric dimension of a graph, Ars Comb. 2 (191–195) (1976) 1.
[12] M. Hauptmann, R. Schmied, C. Viehmann, Approximation complexity of metric dimension problem, J. Discret. Algorithms 14 (2012) 214–222, Selected papers from the 21st International Workshop on Combinatorial Algorithms (IWOCA 2010).
[13] Z. Jiang, N. Polyanskii, On the metric dimension of Cartesian powers of a graph, J. Comb. Theory, Ser. A 165 (2019) 1–14.
[14] S. Khuller, B. Raghavachari, A. Rosenfeld, Landmarks in graphs, Discrete Appl. Math. 70 (3) (1996) 217–229.
[15] L. Laird, R.C. Tillquist, S. Becker, M.E. Lladser, Resolvability of Hamming graphs, SIAM J. Discrete Math. 34 (4) (2020) 2063–2081.
[16] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet Physics Doklady, vol. 10, 1966, pp. 707–710.
[17] V.I. Levenshtein, Efficient reconstruction of sequences from their subsequences or supersequences, J. Comb. Theory, Ser. A 93 (2) (Feb. 2001) 310–332.
[18] V. Mäkinen, E. Ukkonen, G. Navarro, Approximate matching of run-length compressed strings, Algorithmica 35 (4) (2003) 347–369.
[19] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48 (3) (1970) 443–453.
[20] N. Pisanti, Recent Duplications in Genomes: A Graph Theory Approach, Université de Marne-la-Vallée, 1998, DEA memoire.
[21] P. Ruth, Numerical Encoding of Symbolic Data: Standard, State of the Art, and New Techniques, Undergraduate Honors Thesis, University of Colorado, March 2021.
[22] F. Sala, R. Gabrys, C. Schoeny, L. Dolecek, Three novel combinatorial theorems for the insertion/deletion channel, in: 2015 IEEE International Symposium on Information Theory (ISIT), IEEE, 2015, pp. 2702–2706.
[23] P.J. Slater, Leaves of trees, Congr. Numer. 14 (549–559) (1975) 37.
[24] F. Stahlberg, Discovering vocabulary of a language through cross-lingual alignment, PhD thesis, Karlsruhe Institute of Technology, 2011.
[25] R.C. Tillquist, R.M. Frongillo, M.E. Lladser, Metric dimension, Scholarpedia 14 (10) (2019) 53881, revision #190769.
[26] R.C. Tillquist, R.M. Frongillo, M.E. Lladser, Getting the lay of the land in discrete space: A survey of metric dimension and its applications, to appear in SIAM Rev. (2021).
[27] R.C. Tillquist, M.E. Lladser, Low-dimensional representation of genomic sequences, J. Math. Biol. 79 (1) (2019) 1–29, p. 7.
[28] E. Ukkonen, Algorithms for approximate string matching, Inf. Control 64 (1–3) (1985) 100–118.
[29] L.R. Varshney, J. Kusuma, V.K. Goyal, On palimpsests in neural memory: an information theory viewpoint, IEEE Trans. Molec. Biol. Multi-Scale Commun. 2 (2) (2016) 143–153, p. 12.
[30] R.A. Wagner, M.J. Fischer, The string-to-string correction problem, J. ACM 21 (1) (1974) 168–173.
[31] X. Zhong, F. Heinicke, S. Rayner, miRBaseMiner, a tool for investigating miRBase content, RNA Biol. 16 (11) (2019) 1534–1546.