The Linear Complexity of a Graph

David L. Neel
Department of Mathematics
Seattle University, Seattle, WA, USA
neeld@seattleu.edu

Michael E. Orrison
Department of Mathematics
Harvey Mudd College, Claremont, CA, USA
orrison@hmc.edu

Submitted: Aug 1, 2005; Accepted: Jan 18, 2006; Published: Feb 1, 2006
Mathematics Subject Classification: 05C85, 68R10

Abstract

The linear complexity of a matrix is a measure of the number of additions, subtractions, and scalar multiplications required to multiply that matrix and an arbitrary vector. In this paper, we define the linear complexity of a graph to be the linear complexity of any one of its associated adjacency matrices. We then compute or give upper bounds for the linear complexity of several classes of graphs.

1 Introduction

Complexity, like beauty, is in the eye of the beholder. It should therefore come as no surprise that the complexity of a graph has been measured in several different ways. For example, the complexity of a graph has been defined to be the number of its spanning trees [2, 5, 8]. It has been defined to be the value of a certain formula involving the number of vertices, edges, and proper paths in a graph [10]. It has also been defined as the number of Boolean operations, based on a pre-determined set of Boolean operators (usually union and intersection), necessary to construct the graph from a fixed generating set of graphs [12].

In this paper, we introduce another measure of the complexity of a graph. Our measure is the linear complexity of any one of the graph's adjacency matrices. If A is any matrix, then the linear complexity of A is essentially the minimum number of additions, subtractions, and scalar multiplications required to compute AX, where X is an arbitrary column vector of the appropriate size [4]. As we will see, all of the adjacency matrices of a graph Γ have the same linear complexity. We define this common value to be the *linear complexity* of Γ (see Sections 2.2-2.3).

An adjacency matrix of a graph completely encodes its underlying structure. Moreover, this structure is completely recoverable using any algorithm designed to compute the product of an adjacency matrix of the graph and an arbitrary vector. The linear complexity of a graph may therefore be seen as a measure of its overall complexity in that it measures our ability to efficiently encode its adjacency matrices. In other words, it measures the ease with which we are able to communicate the underlying structure of a graph.

Our original motivation for studying the linear complexity of a graph was the fact that the number of arithmetic operations required to compute the projections of an arbitrary vector onto the eigenspaces of an adjacency matrix can be bounded using its size, number of distinct eigenvalues, and linear complexity [9, 11]. Knowing the linear complexity of a graph therefore gives us some insight into how efficiently we can compute certain eigenspace projections. Such insights can be extremely useful when computing, for example, amplitude spectra for fitness functions defined on graphs (see, for example, [1, 13, 14]).

The linear complexities of several classes of matrices, including discrete Fourier transforms, Toeplitz, Hankel, and circulant matrices, have been studied [4]. Since our focus is on the adjacency matrices of graphs, this paper may be seen as contributing to the understanding of the linear complexity of the class of symmetric 0–1 matrices. For example, with only slight changes, many of our results carry over easily to symmetric 0–1 matrices by simply allowing graphs to have loops.

We proceed as follows. In Section 2, we describe the linear complexity of a matrix, and we introduce the notion of the linear complexity of a graph. We also see how we may relate the linear complexity of a graph to that of one of its subgraphs. In Section 3, we give several upper and lower bounds on the linear complexity of a graph. In Section 4, we consider the linear complexity of several well-known classes of graphs. Finally, in Section 5, we give an upper bound for the linear complexity of a graph that is based on the use of clique partitions.

2 Background

In this section, we define the linear complexity of a graph. Our approach requires only a basic familiarity with adjacency matrices of graphs, and a working knowledge of the linear complexity of a linear transformation. An excellent reference for linear complexity, and algebraic complexity in general, is [4]. Throughout the paper, we assume familiarity with the basics of graph theory. See for example [15]. Lastly, all graphs considered in this paper are finite, simple, and undirected.

2.1 Adjacency Matrices

Let Γ be a graph whose vertex set is $\{\gamma_1, \ldots, \gamma_n\}$. The corresponding *adjacency matrix* of Γ is the symmetric $n \times n$ matrix whose (i, j) entry is 1 if γ_i is adjacent to γ_j, and 0 otherwise. For example, if Γ is the complete graph on four vertices (see Figure 1), then its adjacency matrix is

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

regardless of the order of its vertices. If Γ is a cycle on four vertices (see Figure 1), then it has three distinct adjacency matrices:

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

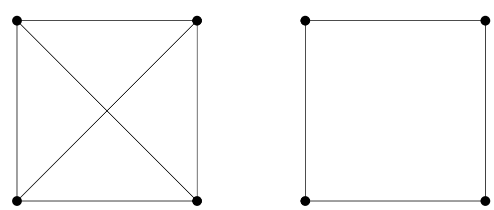

Figure 1: A complete graph (left) and cycle (right) on four vertices.

Note that, for convenience, we will often speak of "the" adjacency matrix of Γ when it is clear that a specific choice of an ordering of the vertices of Γ is inconsequential.

2.2 Linear Complexity of a Matrix

Let K be a field and let

$$(g_{-n+1}, \ldots, g_0, g_1, \ldots, g_r)$$

be a sequence of linear forms in indeterminants $x_1, \ldots, x_n$ over K (i.e., linear combinations of the x_i with coefficients in K). As defined in [4], such a sequence is a *linear computation sequence* (over K with n inputs) *of length* r if

1. $g_{-n+1} = x_1, \ldots, g_0 = x_n$ and,
2. for every $1 \leq \rho \leq r$, either

   $$g_\rho = z_\rho g_i \text{ or } g_\rho = \epsilon_\rho g_i + \delta_\rho g_j,$$

   where $0 \neq z_\rho \in K$, $\epsilon_\rho, \delta_\rho \in \{+1, -1\}$, and $-n < i, j < \rho$.

Such a sequence is then said to *compute* a set F of linear forms if F is a subset of $\{0, \pm g_\rho \mid -n < \rho \leq r\}$.

As an example, if $K = \mathbb{R}$, $F = \{x_1 + x_2, x_1 - 3x_2\}$, and $F' = \{x_1 + x_2, 2x_1 - 2x_3, 4x_1 + 2x_3\}$, then

$$(x_1, x_2, x_1 + x_2, 3x_2, x_1 - 3x_2)$$

is a linear computation sequence of length 3 that computes F, and

$$(x_1, x_2, x_3, x_1 + x_2, 2x_1, 2x_3, 2x_1 - 2x_3, 4x_1, 4x_1 + 2x_3)$$

is a linear computation sequence of length 6 that computes F'.

The *linear complexity* $L(f_1, \ldots, f_m)$ of the set $\{f_1, \ldots, f_m\}$ of linear forms is the minimum $r \in \mathbb{N}$ such that there is a linear computation sequence of length r that computes $\{f_1, \ldots, f_m\}$. The *linear complexity* $L(A)$ of a matrix $A = (a_{ij}) \in K^{m \times n}$ is then defined to be $L(f_1, \ldots, f_m)$, where $f_i = \sum_{j=1}^n a_{ij} x_j$. The linear complexity of a matrix $A \in K^{m \times n}$ is therefore a measure of how difficult it is to compute the produt AX, where $X = [x_1, \ldots, x_n]^t$ is an arbitrary vector.

Note that, for convenience, we will assume that all of the linear computation sequences in this paper are over a field K of characteristic 0. Also, before moving on to graphs, we list here as lemmas some linear complexity results that will be useful in later sections:

Lemma 1 (Remark 13.3 (4) in [4]). *Let $\{f_1, \ldots, f_m\}$ be a set of linear forms in the variables $x_1, \ldots, x_n$. If $\{f_1, \ldots, f_m\} \cap \{0, \pm x_1, \ldots, \pm x_n\} = \emptyset$ and $f_i \neq f_j$ for all $i \neq j$, then $L(f_1, \ldots, f_n) \geq m$.*

Lemma 2 (Lemma 13.7 (2) in [4]). *If B is a submatrix of A, i.e., $B = A$ or B is obtained from A by deleting some rows and/or columns, then $L(B) \leq L(A)$.*

Lemma 3 (Corollary 13.21 in [4]). *$L(\sum_{i=1}^n a_i x_i) = n - 1 + |\{|a_1|, \ldots, |a_n|\} \setminus \{1\}|$, if all of the a_i are nonzero.*

2.3 Linear Complexity of a Graph

Let Γ be a graph, let $\{\gamma_1, \ldots, \gamma_n\}$ be its vertex set, and let $A = (a_{ij}) \in \{0,1\}^{n \times n}$ be its associated adjacency matrix. To every vertex $\gamma_i \in \Gamma$, we will associate the indeterminant x_i and the linear form

$$f_i = \sum_{j=1}^n a_{ij} x_j.$$

Since $a_{ij} = 1$ if $\gamma_i \sim \gamma_j$ and is 0 otherwise, f_i depends only on the neighbors of γ_i. In particular, it should be clear that $L(f_i) \leq \deg(\gamma_i) - 1$.

As we have seen, different orderings of the vertices of a graph give rise to possibly different adjacency matrices. Since the linear forms of different adjacency matrices of a graph differ only by a permutation, however, we may unambiguously define the *linear complexity* $L(\Gamma)$ of a graph Γ to be the linear complexity of any one of its adjacency matrices. In other words, the linear complexity of a graph Γ is a measure of how hard it is to compute AX, where A is an adjacency matrix of Γ and X is a generic vector of the appropriate size.

2.4 Reduced Version of a Matrix

We now turn our attention to relating the linear complexity of one matrix to another. We begin with the following theorem which relates the linear complexity of a matrix to that of its transpose. It is a slightly modified version of Theorem 13.20 in [4], and it will play a pivotal role in the next section and, consequently, throughout the rest of the paper.

Theorem 4 (Theorem 13.20 in [4]). *If $z(A)$ denotes the number of zero rows of $A \in K^{m\times n}$, then*

$$L(A) = L(A^t) + n - m + z(A) - z(A^t).$$

If A is a matrix and B is obtained from A by removing redundant rows, rows of zeros, and rows that contain all zeros except for a single one, then $L(A) = L(B)$. Such rows will contribute nothing to the length of any linear computation sequence of A since they contribute no additional linear forms. We will call this matrix the *reduced version* of A and will denote it by $r(A)$. For our purposes, the usefulness of Theorem 4 lies in our ability to relate $L(A) = L(r(A))$ to $L(r(A)^t)$. Furthermore, we may do this recursively.

As an example, if

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1}$$

then

$$r(A) = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

since the third and sixth rows of A are equal to the first and fifth rows of A, respectively, and the seventh row contains all zeros except for one 1. The reduced version of the transpose of $r(A)$ is

$$r(r(A)^t) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

and the reduced version of the transpose of $r(r(A)^t)$ is

$$r(r(r(A)^t)^t) = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}. \tag{2}$$

By using these reduced matrices, and repeatedly appealing to Theorem 4, we see that

$$\begin{aligned} L(A) &= L(r(A)) = L(r(A)^t) + 3 \\ &= L(r(r(A)^t)) + 3 \\ &= L(r(r(r(A)^t)^t)) + 4 \\ &= 7 \end{aligned}$$

since it can be shown that the matrix $r(r(r(A)^t)^t)$ in (2) has linear complexity 3 (see, for example, Theorem 17).

2.5 Irreducible Graphs

To see the above discussion from a graph-theoretic perspective, consider the graph corresponding to the matrix in (1) (see Figure 2). In this case, we see that the neighbor sets of γ_1 and γ_3 are equal, as are the neighbor sets of γ_5 and γ_6. In addition, γ_7 is a leaf. If we remove γ_3, γ_6 and γ_7, then γ_5 becomes a leaf. By then removing γ_5, we leave only a cycle on $\gamma_1, \gamma_2, \gamma_4$. Using Theorem 4, we may then relate the linear complexities of these subgraphs.

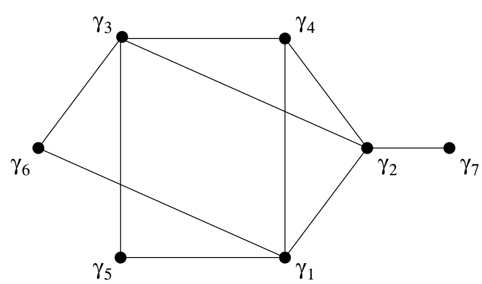

Figure 2: A reducible graph.

To make this idea concrete, consider constructing a sequence of connected subgraphs of a connected graph in the following way. First, if γ is a vertex in a graph Γ, then we will denote its neighbor set in Γ by $N_\Gamma(\gamma)$. Also, if Γ is a graph, then $V(\Gamma)$ will denote its vertex set, and $E(\Gamma)$ will denote its edge set.

Let Γ be a connected graph with vertex set $V(\Gamma) \subseteq \{\gamma_1, \ldots, \gamma_n\}$ consisting of at least three vertices. Let $R(\Gamma)$ denote the subgraph of Γ obtained by removing the vertex $\gamma_j \in V(\Gamma)$ with the smallest index j such that

1. γ_j is a leaf, or

2. there exists a $\gamma_i \in V(\Gamma)$ such that $i < j$ and $N_\Gamma(\gamma_j) = N_\Gamma(\gamma_i)$.

If no such vertex exists, then define $R(\Gamma)$ to be Γ. For convenience, we also define $R(\Gamma)$ to be Γ if Γ consists of only one edge or one vertex. If Γ is a connected graph such that $R(\Gamma) = \Gamma$, then we say that Γ is *irreducible*. If $R(\Gamma) \neq \Gamma$, then we say that Γ is *reducible*.

Given a connected graph Γ with vertex set $V(\Gamma) = \{\gamma_1, \ldots, \gamma_n\}$, we may then construct the sequence

$$\Gamma, R(\Gamma), R(R(\Gamma)), R(R(R(\Gamma))), \ldots$$

of subgraphs of Γ. Let $I(\Gamma)$ denote the first irreducible graph in this sequence.

Theorem 5. *If Γ is a connected graph Γ with vertex set $V(\Gamma) = \{\gamma_1, \ldots, \gamma_n\}$, then*

$$L(\Gamma) = L(I(\Gamma)) + |V(\Gamma)| - |V(I(\Gamma))|.$$

Proof. It suffices to show that if Γ is not irreducible, then $L(\Gamma) = L(R(\Gamma))+1$. With that in mind, suppose Γ is not irreducible, and let $\gamma \in V(G)$ be the vertex removed from Γ to create $R(\Gamma)$. Let A be the adjacency matrix of Γ, and let B be the matrix obtained from A by removing the row corresponding to γ. By construction, we know that $L(A) = L(B)$. By Theorem 4, we then have that

$$L(B) = L(B^t) + 1.$$

By removing the redundant row in B^t that corresponds to γ, we then create the adjacency matrix A' of $R(\Gamma)$. Moreover, since $L(B^t) = L(A')$, we have that $L(A) = L(A') + 1$. In other words, $L(\Gamma) = L(R(\Gamma)) + 1$. $\square$

3 Bounds

In this section, we consider bounds on the linear complexity of a graph. We begin with some naive bounds. We then consider bounds based on edge partitions and direct products.

3.1 Naive Bounds

We begin with some naive but useful bounds on the linear complexity of a graph.

Proposition 6. *If Γ is a connected graph, then $L(\Gamma) \leq 2|E(\Gamma)| - |V(\Gamma)|$.*

Proof. The linear form associated to each $\gamma \in V(\Gamma)$ requires at most $\deg(\gamma) - 1 \geq 0$ additions. Thus,

$$L(\Gamma) \leq \sum_{\gamma \in V(\Gamma)} (\deg(\gamma) - 1) = 2|E(\Gamma)| - |V(\Gamma)|.$$

$\square$

Since the linear form associated to a vertex depends only on its neighbors, we also have the following bound.

Proposition 7. *The linear complexity of a graph is less than or equal to the sum of the linear complexities of its connected components.*

Proposition 8. *If Γ is a graph and $\Delta(\Gamma)$ is the maximum degree of a vertex in Γ, then $\Delta(\Gamma) - 1 \leq L(\Gamma)$.*

Proof. Let A be the adjacency matrix of Γ. Remove all of the rows of A except for one row corresponding to a vertex of maximum degree, and call the resulting row matrix B. By Lemma 2, we have that $L(B) \leq L(A)$, and by Lemma 3, we have that $L(B) = \Delta(\Gamma) - 1$. The proposition follows immediately. □

We may put an equivalence relation on the vertices of a graph Γ by saying that two vertices are equivalent if they have precisely the same set of neighbors. Since equivalent vertices are never adjacent, note that each equivalence class is an independent set of vertices.

Proposition 9. *Let Γ be a connected graph. If m is the number of equivalence classes (of the equivalence relation defined above) that contain non-leaves, then $m \leq L(\Gamma)$.*

Proof. Let A be the adjacency matrix of Γ. The nontrivial linear forms of A correspond to the non-leaves of Γ. Since equivalent vertices have equal linear forms, we need only consider m distinct nontrivial linear forms. The proposition then follows immediately from Lemma 1. □

Although Proposition 9 is indeed a naive bound, it suggests that we may find minimal linear computation sequences for the adjacency matrix of a graph by gathering together vertices whose corresponding linear forms are equal, or nearly equal. This approach will be particularly useful when we consider complete graphs and complete k-partite graphs in Section 4.

3.2 Bounds from Partitioning Edge Sets

We now consider upper bounds on the linear complexity of a graph obtained from a partition of its edge set.

Theorem 10. *Let Γ be a graph and suppose that $E(\Gamma)$ is the union of k disjoint subsets of edges such that the jth subset induces the subgraph Γ_j of Γ. If Γ has n vertices and the ith vertex is in b_i of the induced subgraphs, then*

$$L(\Gamma) \leq \sum_{j=1}^{k} L(\Gamma_j) + \sum_{i=1}^{n} (b_i - 1).$$

Proof. Let $V(\Gamma) = \{\gamma_1, \ldots, \gamma_n\}$ be the vertex set of Γ. As noted in Section 2.3, we may assume that to $\gamma_i \in V(\Gamma)$ we have associated the indeterminant x_i and the linear form

$$f_i = \sum_{j=1}^{n} a_{ij} x_j$$

where $a_{ij} = 1$ if $\gamma_i \sim \gamma_j$ and is 0 otherwise. If $\gamma_i \in \Gamma_j$, then let f_i^j be the linear form associated to γ_i when thought of as a vertex in Γ_j. If $\gamma_i \notin \Gamma_j$, define $f_i^j = 0$. It follows that

$$f_i = \sum_{j=1}^{k} f_i^j.$$

This sum has b_i nonzero summands, so its linear complexity is no more than $b_i - 1$ if given the linear forms $f_i^1, \ldots, f_i^k$. Since the linear complexity of the set of f_i^j is no more than $\sum_{j=1}^{k} L(\Gamma_j)$, the theorem follows. $\square$

In the last theorem, we saw how the linear complexity of a graph can be bounded by the linear complexity of edge-disjoint subgraphs. In the next theorem, we consider the linear complexity of a graph obtained by removing edges from another graph.

Let Γ be a graph and let $F \subseteq E(\Gamma)$. Let Γ_F be the subgraph of Γ induced by F and let $\Gamma_{\widetilde{F}}$ be the subgraph of Γ obtained by removing the edges in F. Finally, let $\overline{F}$ be the complement of F in $E(\Gamma)$.

Theorem 11. *If Γ is a graph and $F \subseteq E(\Gamma)$, then*

$$L(\Gamma_{\widetilde{F}}) \leq L(\Gamma) + L(\Gamma_F) + |V(\Gamma_F) \cap V(\Gamma_{\overline{F}})|.$$

Proof. Let $V(\Gamma) = \{\gamma_1, \ldots, \gamma_n\}$ be the vertex set of Γ. To $\gamma_i \in V(\Gamma)$, associate the indeterminant x_i and the linear form

$$f_i = \sum_{j=1}^{n} a_{ij} x_j$$

where $a_{ij} = 1$ if $\gamma_i \sim \gamma_j$ and is 0 otherwise. Let $f_i^{\widetilde{F}}$ be the linear form associated to γ_i as a vertex in $\Gamma_{\widetilde{F}}$. If $\gamma_i \in \Gamma_F$, then let f_i^F be the linear form associated to γ_i as a vertex in Γ_F. If $\gamma_i \notin \Gamma_F$, define $f_i^F = 0$. It follows that

$$f_i^{\widetilde{F}} = f_i - f_i^F.$$

This difference is nontrivial only if $\gamma_i \in V(\Gamma_F) \cap V(\Gamma_{\overline{F}})$. Since the linear complexity of $\{f_i\}_{i=1}^n \cup \{f_i^F\}_{i=1}^n$ is no more than $L(\Gamma) + L(\Gamma_F)$, the theorem follows. $\square$

The next theorem gives a bound for the difference between the complexity of a graph and that of a subgraph induced by a subset of edges. Our proof relies on Theorem 10, Theorem 11, and the following lemma.

Lemma 12. *If Γ is a graph and $F \subseteq E(\Gamma)$, then $L(\Gamma_{\widetilde{F}}) = L(\Gamma_{\overline{F}})$.*

Proof. $\Gamma_{\widetilde{F}}$ is isomorphic to $\Gamma_{\overline{F}}$ together with a (possibly empty) set of isolated vertices. It follows that the sets of nontrivial linear forms associated to $\Gamma_{\widetilde{F}}$ and $\Gamma_{\overline{F}}$ are identical. $\square$

Theorem 13. *If* Γ *is a graph and* $F \subseteq E(\Gamma)$, *then*

$$|L(\Gamma) - L(\Gamma_{\widetilde{F}})| = |L(\Gamma) - L(\Gamma_{\overline{F}})| \leq L(\Gamma_F) + |V(\Gamma_F) \cap V(\Gamma_{\overline{F}})|.$$

Proof. The edge set of Γ is the disjoint union of F and $\overline{F}$. Thus, by Theorem 10 we have

$$L(\Gamma) - L(\Gamma_{\overline{F}}) \leq L(\Gamma_F) + |V(\Gamma_F) \cap V(\Gamma_{\overline{F}})|.$$

By Theorem 11 we have

$$L(\Gamma_{\widetilde{F}}) - L(\Gamma) \leq L(\Gamma_F) + |V(\Gamma_F) \cap V(\Gamma_{\overline{F}})|.$$

By Lemma 12, $L(\Gamma_{\widetilde{F}}) = L(\Gamma_{\overline{F}})$. The theorem follows immediately. □

Corollary 14. *If two graphs* Γ *and* Γ' *differ by only one edge, then*

$$|L(\Gamma) - L(\Gamma')| \leq 2.$$

3.3 Bounds for Direct Products of Graphs

Before moving on to specific examples, we finish this section by considering the linear complexity of a graph that is the direct product of other graphs. Examples of such graphs include the important class of Hamming graphs (see Section 4.6).

The *direct product* of d graphs $\Gamma_1, \ldots, \Gamma_d$ is the graph with vertex set $V(\Gamma_1) \times \cdots \times V(\Gamma_d)$ whose edges are the two-element sets $\{(\gamma_1, \ldots, \gamma_d), (\gamma'_1, \ldots, \gamma'_d)\}$ for which there is some m such that $\gamma_m \sim \gamma'_m$ and $\gamma_l = \gamma'_l$ for all $l \neq m$ (see, for example, [3]).

Theorem 15. *If* Γ *is the direct product of* $\Gamma_1, \ldots, \Gamma_d$, *then*

$$L(\Gamma) \leq |V(\Gamma)| \left(\sum_{j=1}^{d} \frac{L(\Gamma_j)}{|V(\Gamma_j)|} + (d-1) \right).$$

Proof. For $1 \leq i \leq d$, let E_i be the subset of edges of Γ whose vertices differ in the ith position, and let Γ_{E_i} be the subgraph of Γ induced by E_i. Note that the E_i partition $E(\Gamma)$ and that Γ_{E_i} is isomorphic to a graph consisting of $\prod_{j \neq i} |V(\Gamma_j)|$ disconnected copies of Γ_i. Since every vertex of Γ is contained in at most d of the E_i, and $|V(\Gamma)| = \prod_{i=1}^{d} |V(\Gamma_i)|$, by Proposition 7 and Theorem 10 we have

$$\begin{aligned}
L(\Gamma) &\leq \sum_{i=1}^{d} L(\Gamma_{E_i}) + |V(\Gamma)|(d-1) \\
&= \sum_{i=1}^{d} \left(\prod_{j \neq i} |V(\Gamma_j)| L(\Gamma_i) \right) + |V(\Gamma)|(d-1) \\
&= \left(\prod_{j=1}^{d} |V(\Gamma_j)| \right) \left(\sum_{i=1}^{d} \frac{L(\Gamma_i)}{|V(\Gamma_i)|} \right) + |V(\Gamma)|(d-1) \\
&= |V(\Gamma)| \left(\sum_{i=1}^{d} \frac{L(\Gamma_i)}{|V(\Gamma_i)|} + (d-1) \right).
\end{aligned}$$

□

Corollary 16. *If Γ^d is the direct product of the graph Γ with itself d times, then*

$$L(\Gamma^d) \leq |V(\Gamma)|^d \left(d\frac{L(\Gamma)}{|V(\Gamma)|} + (d-1) \right).$$

4 Examples

In this section, we consider the linear complexity of several well-known classes of graphs. More specifically, we determine the linear complexity of cycles, trees, complete graphs, and complete k-partitite graphs. We also present bounds on the linear complexity of Johnson graphs and Hamming graphs.

4.1 Cycles

Let C_n denote the graph that is the cycle on n vertices. Since C_n has n edges and n vertices, the naive bound given by Proposition 6 is

$$L(C_n) \leq 2n - n = n.$$

For most cycles, this bound is optimal.

Theorem 17. *If $n \neq 4$, then $L(C_n) = n$.*

Proof. If $n \neq 4$, then every vertex of C_n is a non-leaf with a unique set of neighbors. Thus, by Proposition 9, $n \leq L(C_n)$. Since $L(C_n) \leq n$ by Proposition 6, the theorem follows. □

Note that the vertices of C_4 have non-distinct neighbor sets. In fact, C_4 is isomorphic to the complete bipartite graph $K_{2,2}$ which we consider in Section 4.4.

4.2 Trees

Let T be a tree on $n \geq 2$ vertices. Since T has $n-1$ edges, the naive bound for $L(T)$ given by Proposition 6 is

$$L(T) \leq 2(n-1) - n = n-2.$$

Moreover, this bound is optimal.

Theorem 18. *If T is a tree on $n \geq 2$ vertices, then $L(T) = n-2$.*

Proof. In a tree, only the leaves may have non-distinct neighbor sets, otherwise the tree would contain a cycle. Thus $R(T)$ is again a tree, and $I(T)$ consists of only an edge, which has linear complexity 0. The claim then follows from Theorem 5. □

4.3 Complete Graphs

Recall that K_n denotes the complete graph on n vertices. Since K_n has $n(n-1)/2$ edges, if $n \geq 2$, then the bound given by Proposition 6 is

$$L(K_n) \leq 2\frac{n(n-1)}{2} - n = n(n-2). \tag{3}$$

The adjacency matrix for K_n, however, is quite simple to describe—it has zeros on the diagonal and ones in every other postition. For example, the adjacency matrix for K_5 is

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

We may easily take advantage of this simple structure to create a linear computation sequence for the adjacency matrix of K_n with length much shorter than $n(n-2)$.

Let $\{\gamma_1, \ldots, \gamma_n\}$ be the vertices of K_n. Note that the linear form associated to γ_j is

$$f_j = \left(\sum_{i=1}^{n} x_i\right) - x_j.$$

We may therefore compute $\{f_1, \ldots, f_n\}$ by computing

i. $f_n = \sum_{i=1}^{n-1} x_i$,

ii. $S = \sum_{i=1}^{n} x_i = f_n + x_n$, and

iii. $f_j = S - x_j$ for $j = 1, \ldots, (n-1)$.

These three steps give rise to a linear computation sequence for the f_i of length

$$(n-2) + 1 + (n-1) = 2n - 2.$$

Thus, $L(K_n) \leq 2n - 2$. As we now will show, if $n \geq 4$, then this inequality is in fact equality. We begin with two lemmas.

Lemma 19. *If $n \geq 2$, then $L(K_n) \leq L(K_{n+1}) - 2$.*

Proof. First, note that the adjacency matrix of K_n is a submatrix of the adjacency matrix for K_{n+1}. Thus, by Lemma 2, $L(K_n) \leq L(K_{n+1})$. Let

$$s = (x_1, \ldots, x_{n+1}, g_1, \ldots, g_m)$$

be a minimal linear computation sequence for the adjacency matrix of K_{n+1}. Since $n+1 \geq 3$, we know that $m \geq 3$ by Proposition 8 and Theorem 17.

Since the computation sequence s is minimal, we know that g_m must be a linear form associated to one of the vertices in K_{n+1}. Let γ be this vertex, and let x be the indeterminate associated to γ. We may create a linear computation sequence s' for $K_{n+1} - \gamma \cong K_n$ from s by removing g_m, setting $x = 0$, and removing any redundant linear forms in s. Since setting $x = 0$ makes at least one of the g_i, $1 \leq i \leq m-1$ redundant, we know that the length of s' is at most $m-2$. Thus $L(K_n) \leq L(K_{n+1}) - 2$. □

Lemma 20. $L(K_4) = 6$.

Proof. We know that $L(K_4) \leq 6$, and by Proposition 17 and Lemma 19, we also know that $L(K_4) \geq 5$. We therefore need only show that $L(K_4) \neq 5$.

Suppose that $(x_1, x_2, x_3, x_4, g_1, g_2, g_3, g_4, g_5)$ is a linear computation sequence for the adjacency matrix A of K_4. Since $g_2, \ldots, g_5$ must be (up to sign) the four linear forms associated with A, we may assume without loss of generality that $g_1 = \pm(x_1 + x_2)$ and $g_2 = \pm(x_1 + x_2 + x_3)$. This then forces g_3 to be $\pm(x_1 + x_2 + x_4)$.

Suppose now that $g_4 = \pm(x_1 + x_3 + x_4)$. Since x_1 and x_2 have different coefficients in g_4, it cannot be the case that it is the sum or difference of two forms in $\{g_1, g_2, g_3\}$. Furthermore, any other sum or difference of two forms in $\{x_1, x_2, x_3, x_4, g_1, g_2, g_3\}$ has an even number of nonzero coefficients. Since g_4 is not a multiple of any of the already computed forms, it follows that it is impossible for g_4 to equal $\pm(x_1 + x_3 + x_4)$. A similar argument holds for the form $\pm(x_2 + x_3 + x_4)$. This shows that a linear computation sequence for the adjacency matrix of K_4 must have length at least six. Thus $L(K_4) \neq 5$. □

We may now determine the linear complexity of all complete graphs.

Theorem 21. *$L(K_1) = L(K_2) = 0$, $L(K_3) = 3$, and if $n \geq 4$, then $L(K_n) = 2n - 2$.*

Proof. The linear forms associated to K_1 and K_2 are trivial, thus $L(K_1) = L(K_2) = 0$. Since K_3 is a cycle on three vertices, $L(K_3) = 3$ follows from Theorem 17. For $n \geq 4$, we proceed by induction using $L(K_4) = 6$ as our base case.

Suppose the statement holds for $k \geq 4$, and thus we know that $L(K_k) = 2k - 2$. We have constructed a linear computation sequence above that shows that

$$L(K_{k+1}) \leq 2(k+1) - 2 = 2k.$$

By Lemma 19, we have that $L(K_k) + 2 \leq L(K_{k+1})$. It follows that $L(K_{k+1}) = 2k$. Thus, by induction, if $n \geq 4$, then $L(K_n) = 2n - 2$. □

The following corollary states that the difference between the linear complexity of a graph on n vertices and the linear complexity of its complement is bounded by $3n - 2$. It follows from Lemma 12, Theorem 11, and Theorem 21 although we provide a much more direct proof using adjacency matrices.

Corollary 22. *If Γ is a graph on n vertices, then*

$$|L(\Gamma) - L\left(\overline{\Gamma}\right)| \leq 3n - 2.$$

Proof. Let Γ be a graph on n vertices and let $A(\Gamma)$ denote the adjacency matrix of Γ. We clearly have

$$A(\Gamma) = A(K_n) - A\left(\overline{\Gamma}\right).$$

It follows that

$$L(\Gamma) \leq L(K_n) + L(\overline{\Gamma}) + n = 3n - 2 + L(\overline{\Gamma}).$$

The corollary then follows immediately from the fact that the complement of $\overline{\Gamma}$ is Γ. $\square$

4.4 Complete k-partitite graphs

Theorem 21 generalizes easily to complete k-partite graphs. First, recall that the *complete k-partite graph* $K_{n_1,\dots,n_k}$ is the graph whose vertex set may be partitioned into k blocks of sizes $n_1, \dots, n_k$ such that two vertices are adjacent if and only if they are in different blocks. Note that $K_n = K_{1,\dots,1}$, and that if $k = 2$, then such a graph is also known as a *complete bipartite graph* or a *bipartite clique*. Bipartite cliques will play an important role for us in Section 5.1.

We assume that $k \geq 2$. Let $\{\gamma_1, \dots, \gamma_n\}$ be the vertices of $K_{n_1,\dots,n_k}$ (where $n = n_1 + \cdots + n_k$), and let $B_1, \dots, B_k$ be the corresponding blocks of vertices in the associated partition of $V(K_{n_1,\dots,n_k})$. For $j = 1, \dots, k$, define

$$y_j = \sum_{\gamma_i \in B_j} x_i.$$

Note that if $\gamma_i \in B_j$, then the linear form associated to γ_i is

$$f_j = \left(\sum_{l=1}^{k} y_l\right) - y_j.$$

We therefore need only compute $\{f_1, \dots, f_k\}$ to compute all of the linear forms for $K_{n_1,\dots,n_k}$. This may be done by computing

i. $y_j = \sum_{\gamma_i \in B_j} x_i$ for $j = 1, \dots, k$,

ii. $f_k = \sum_{j=1}^{k-1} y_j$,

iii. $S = \sum_{j=1}^{k} y_j = f_k + y_k$, and

iv. $f_j = S - y_j$ for $j = 1, \dots, (k-1)$.

These four steps give rise to a linear computation sequence for $\{f_1, \dots, f_k\}$ of length

$$(n - k) + (k - 2) + 1 + (k - 1) = n + k - 2.$$

This proves that $L(K_{n_1,\dots,n_k}) \leq n + k - 2$. Moreoever, if $k \geq 4$, then this bound is optimal.

Theorem 23. *Let Γ be the complete k-partite graph $K_{n_1,\dots,n_k}$ on $n = n_1 + \cdots + n_k$ vertices. If $k = 2$, then $L(\Gamma) = n - 2$; if $k = 3$, then $L(\Gamma) = n$; and if $k \geq 4$, then $L(\Gamma) = n + k - 2$.*

Proof. This follows directly from Theorem 5, Theorem 21, and the fact that

$$I(K_{n_1,\ldots,n_k}) = K_k.$$

□

4.5 Johnson graphs

Let $1 \le k \le n$. The *Johnson graph* $J(n,k)$ is the graph whose vertices are the k-element subsets of $\{1,\ldots,n\}$, where two vertices γ and γ' are adjacent if and only if $|\gamma \cap \gamma'| = k-1$. Functions defined on Johnson graphs arise when considering certain types of committee voting data. Knowing the linear complexity of Johnson graphs therefore tells us something about how efficiently we may analyze such data (see, for example, [6, 9]).

Every vertex of $J(n,k)$ has exactly $k(n-k)$ neighbors. Since $|V(J(n,k))| = \binom{n}{k}$, the number of edges in $J(n,k)$ is

$$\binom{n}{k}\frac{k(n-k)}{2}.$$

The naive upper bound for $L(J(n,k))$ given by Proposition 6 is therefore

$$L(J(n,k)) \le \binom{n}{k}k(n-k) - \binom{n}{k}.$$

We may, however, improve this bound significantly by closely examining the substructure of $J(n,k)$.

Label every edge $\{\gamma,\gamma'\}$ of $J(n,k)$ with the $(k-1)$-element set $\gamma \cap \gamma'$. These labels then partition the edges of $J(n,k)$ into $\binom{n}{k-1}$ subsets, each of which induce a complete subgraph that is isomorphic to K_{n-k+1}. Every vertex of $J(n,k)$ is contained in k such subgraphs. By Theorem 10 and Theorem 21, we therefore have that

$$\begin{aligned} L(J(n,k)) &\le \binom{n}{k-1}(2(n-k+1)-2) + \binom{n}{k}(k-1) \\ &= \binom{n}{k}\left(\frac{2k(n-k)}{n-k+1} + (k-1)\right) \\ &< \binom{n}{k}3k. \end{aligned}$$

This bound, however, may be improved further still.

Theorem 24. *Let $1 \le k \le n$ and let $J(n,k)$ the Johnson graph defined on the k-sets of the set $\{1,\ldots,n\}$. Then*

$$L(J(n,k)) < \binom{n}{k}(2k+1).$$

Proof. Consider the partition of the edges of $J(n,k)$ described above. For every $(k-1)$-element subset δ, let J_δ be the subgraph of $J(n,k)$ that is induced by the edges with the label δ. As noted above, J_δ is a complete graph on $n-k+1$ vertices.

Let $\{\gamma_1,\ldots,\gamma_m\}$ be the vertices of $J(n,k)$. For each J_δ, define

$$S_\delta = \sum_{\gamma_i \in J_\delta} x_i.$$

The linear form associated to any vertex γ_j of $J(n,k)$ is then

$$f_j = \left(\sum_{\delta \subset \gamma_j} S_\delta\right) - kx_j.$$

This immediately gives rise to a linear computation sequence for $\{f_1,\ldots,f_m\}$ of length

$$\binom{n}{k-1}((n-k+1)-1) + (k-1)\binom{n}{k} + 2\binom{n}{k}$$

which simplifies to

$$\binom{n}{k}\frac{k(n-k)}{n-k+1} + (k+1)\binom{n}{k} < \binom{n}{k}(2k+1).$$

□

4.6 Hamming Graphs

A direct product $K_{n_1}\times\cdots\times K_{n_d}$ of complete graphs is a *Hamming graph* (see, for example, [3]). Hamming graphs have recently been used in the analysis of fitness landscapes derived from RNA folding [14]. As with the Johnson graphs and committee voting data, knowing something about the linear complexity of Hamming graphs tells us something about how efficiently we may analyze such landscapes.

Theorem 25. *If Γ is the Hamming graph $K_{n_1}\times\cdots\times K_{n_d}$, then*

$$L(\Gamma) < (2d+1)\left(\prod_{i=1}^{d} n_i\right).$$

Proof. The result follows directly from an argument similar to that found in the proof of Theorem 24. We leave the details to the interested reader. □

In particular, for the Hamming graphs $H(d,n) = K_n\times\cdots\times K_n$ (d times), we have that $L(H(d,n)) < (2d+1)n^d$.

5 Bounds for Graphs in General

In this section, we give an upper bound on the linear complexity of an arbitrary graph. The bound is based on the number of vertices and edges in the graph, and it follows from a result found in [7].

5.1 Clique Partitions

Let Γ be a graph. A *clique partition* of Γ is a collection $\mathcal{C} = \{\Gamma_1, \ldots, \Gamma_k\}$ of subgraphs of Γ such that each Γ_i is a bipartite clique and $\{E(\Gamma_1), \ldots, E(\Gamma_k)\}$ is a partition of $E(\Gamma)$. The *order* of a bipartite clique is the number of vertices in it. The *order* of $\mathcal{C}$ is then defined to be the sum of the orders of the individual Γ_i's.

Let Γ be a graph with n vertices $\{\gamma_1, \ldots, \gamma_n\}$. Let $\mathcal{C} = \{\Gamma_1, \ldots, \Gamma_k\}$ be a clique parition of Γ. Let w_j denote the order Γ_j. By Theorem 23,

$$L(\Gamma_j) = w_j - 2 \tag{4}$$

for each $j = 1, \ldots, k$.

Let $w = w_1 + \cdots + w_k$ be the order of $\mathcal{C}$, and let b_i be the number of Γ_j that contain the ith vertex γ_i of Γ. Note that b_i is then the contribution of γ_i to the order w of $\mathcal{C}$. It follows immediately that

$$\sum_{i=1}^{n} b_i = w. \tag{5}$$

By Theorem 10, (4) and (5) we therefore have that

$$L(\Gamma) \leq \sum_{j=1}^{k} L(\Gamma_j) + \sum_{i=1}^{n} (b_i - 1) = 2w - (n + 2k) < 2w.$$

In other words, the linear complexity of a graph Γ is bounded above by twice the order of any clique partition of Γ. It was shown in [7], however, that if Γ is a graph with n vertices and m edges, then Γ has a clique partition of order

$$O\left(\frac{m \log(n^2/m)}{\log n}\right).$$

This proves the following theorem:

Theorem 26. *If Γ is a graph with n vertices and m edges, then*

$$L(\Gamma) \in O\left(\frac{m \log(n^2/m)}{\log n}\right).$$

6 Acknowledgments

This paper is dedicated to the memory of our friend and mentor Kenneth P. Bogart. Part of this paper was written while the second author was visiting the Department of Mathematics and Statistics at the University of Western Australia. Special thanks to Cheryl Praeger for her encouragement and support. Special thanks also to Doug West for helpful suggestions on terminology and definitions, and to Robert Beezer, Masanori Koyama, Jason Rosenhouse, and Robin Thomas for comments on a preliminary version of this paper. This work was supported in part by a Harvey Mudd College Beckman Research Award and by Seattle University's College of Science and Engineering.

References

[1] O. Bastert, D. Rockmore, P. Stadler, and G. Tinhofer, *Landscapes on spaces of trees*, Appl. Math. Comput. **131** (2002), no. 2-3, 439–459.

[2] N. Biggs, *Algebraic graph theory*, Cambridge University Press, London, 1974, Cambridge Tracts in Mathematics, No. 67.

[3] A. Brouwer, A. Cohen, and A. Neumaier, *Distance-regular graphs*, Springer-Verlag, Berlin, 1989.

[4] P. Bürgisser, M. Clausen, and M.A. Shokrollahi, *Algebraic complexity theory*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 315, Springer-Verlag, Berlin, 1997, With the collaboration of Thomas Lickteig.

[5] G. Constantine, *Graph complexity and the Laplacian matrix in blocked experiments*, Linear and Multilinear Algebra **28** (1990), no. 1-2, 49–56.

[6] P. Diaconis, *Group representations in probability and statistics*, Institute of Mathematical Statistics, Hayward, CA, 1988.

[7] T. Feder and R. Motwani, *Clique partitions, graph compression and speeding-up algorithms*, J. Comput. System Sci. **51** (1995), no. 2, 261–272.

[8] R. Grone and R. Merris, *A bound for the complexity of a simple graph*, Discrete Math. **69** (1988), no. 1, 97–99.

[9] D. Maslen, M. Orrison, and D. Rockmore, *Computing isotypic projections with the Lanczos iteration*, SIAM Journal on Matrix Analysis and Applications **25** (2004), no. 3, 784–803.

[10] D. Minoli, *Combinatorial graph complexity*, Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur. (8) **59** (1975), no. 6, 651–661 (1976).

[11] M. Orrison, *An eigenspace approach to decomposing representations of finite groups*, Ph.D. thesis, Dartmouth College, 2001.

[12] P. Pudlák, V. Rödl, and P. Savický, *Graph complexity*, Acta Inform. **25** (1988), no. 5, 515–535.

[13] C. Reidys and P. Stadler, *Combinatorial landscapes*, SIAM Rev. **44** (2002), no. 1, 3–54.

[14] D. Rockmore, P. Kostelec, W. Hordijk, and P. Stadler, *Fast Fourier transform for fitness landscapes*, Appl. Comput. Harmon. Anal. **12** (2002), no. 1, 57–76.

[15] D. West, *Introduction to graph theory*, Prentice Hall Inc., Upper Saddle River, NJ, 1996.