1
Entropy, Orbits, and Spectra of Graphs

Abbe Mowshowitz and Valia Mitsou

1.1
Introduction

This chapter is concerned with the notion of entropy as applied to graphs for the purpose of measuring complexity.

Most studies of complexity focus on the execution time or space utilization of algorithms. The execution time of an algorithm is proportional to the number of operations required to produce the output as a function of the input size. Space utilization measures the amount of storage required for computation. Both time and space complexity measure the resources required to perform a computation for a specified input. Measuring the complexity of a mathematical object such as a graph is an exercise in structural complexity. This type of complexity does not deal directly with the costs of computation; rather, it offers insight into the internal organization of an object. The structural complexity of a computer program, for example, may indicate the difficulty of modifying or maintaining the program.

One approach to structural complexity involves the length of a code needed to specify an object uniquely (Kolmogorov complexity). The complexity of a string, for example, is the length of the string's shortest description in a given description language [27]. The approach taken in this chapter centers on finding indices of structure, based on Shannon's entropy measure. Unlike Kolmogorov complexity, such an index captures a particular feature of the structure of an object. The symmetry structure of a graph provides the basis for the index explored here.

The choice of symmetry is dictated by its utility in many scientific disciplines. D'Arcy Thompson's classic work [25] showed the relevance of symmetry in the natural world. Structure-preserving transformations based on symmetry play a role in physics, chemistry, and sociology as well as in biology. A symmetry transformation of a graph is typically an edge-preserving bijection of the vertices, i.e., an isomorphism of the graph onto itself. Such a transformation is called an *automorphism*. If the vertices of the graph are labeled, an automorphism can be viewed as a permutation of the vertices that preserves adjacencies. The set of all automorphisms forms a group

Analysis of Complex Networks: From Biology to Linguistics. Edited by Matthias Dehmer and Frank Emmert-Streib
Copyright © 2009 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-32345-6

whose orbits provide the foundation for applying Shannon's entropy measure.

The collection of orbits of the automorphism group constitutes a partition and thus defines an equivalence relation on the vertices of a graph. Two vertices in the same orbit are similar in some sense. In a social network, collections of similar vertices can be used to define communities with shared attributes. The identification of such communities is of interest in applications such as advertising, intelligence, and sensor networks.

Measures of structural complexity are useful for classifying graphs and networks represented by graphs. One is led to conjecture, for example, that the more symmetric a network is (or the lower its automorphism-based complexity is), the more vulnerable to attack it will be. These related issues are explored in [19] in relation to sensor networks modeled as dynamic distributed federated databases [2].

In what follows we define the measure of graph complexity, discuss algorithms and heuristics for computing it, and examine its relationship to another prominent entropy measure [11] defined on graphs.

1.2
Entropy or the Information Content of Graphs

Given a decomposition of the vertices or edges of a graph, one can construct a *finite probability scheme* [10] and compute its entropy. A finite probability scheme assigns a probability to each subset in the decomposition. Such a numerical measure can be seen to capture the information contained in some particular aspect of the graph structure.

The orbits of the automorphism group of a graph constitute a decomposition of the vertices of the graph. As noted above, this decomposition captures the symmetry structure of the graph, and the entropy of the finite probability scheme obtained from the automorphism group provides an index of the complexity of the graph relative to the symmetry structure.

Let $G = (V, E)$ be a graph with vertex set V (with $|V| = n$) and edge set E. The *automorphism group* of G, denoted by $Aut(G)$, is the set of all adjacency-preserving bijections of V. Let $\{V_i | 1 \leq i \leq k\}$ be the collection of orbits of $Aut(G)$, and suppose $|V_i| = n_i$ for $1 \leq i \leq k$. The *entropy* or *information content* of G is given by the following formula ([13]):

$$I_a(G) = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n} .$$

For example, the orbits of the graph of Figure 1.1 are {1}, {2,5}, and {3,4}, so the information content of the graph is $I_a(G) = -\frac{1}{5} \log \frac{1}{5} - 2(\frac{2}{5} \log \frac{2}{5}) = \log 5 - \frac{4}{5} \log 2$.

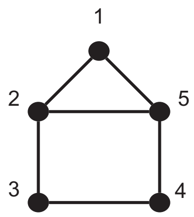


Figure 1.1 Information content of a graph.

Clearly, $I_a(G)$ satisfies $0 \leq I_a(G) \leq \log n$, where the minimum value occurs for graphs with the transitive automorphism group, such as the cycle C_n and the complete graph K_n on n vertices; the maximum is achieved for graphs with the identity group. The smallest nontrivial, undirected graph with an identity group is shown in Figure 1.2.

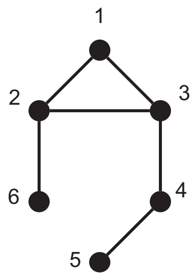


Figure 1.2 Smallest nontrivial graph with identity group.

The idea of measuring the information content of a graph was first presented in [21]; it was formalized in [26] and further developed in [13–16]. $I_a(G)$ is a function of the partition of the vertices of G determined by the orbits of $Aut(G)$. As such the measure captures the structure of vertex similarity. In the case of organic molecules, the lower the information content (or the greater the symmetry), the fewer the possibilities for different interactions with other molecules. If all the atoms are in the same equivalence class, then it makes no difference which one interacts with an atom of another molecule. The same can be said for social networks. Any member of an equivalence class of similar individuals can serve as a representative of the class.

The utility of the measure $I_a(G)$ can be seen from the following special case. The cartesian product $G \times H$ of graphs G and H is defined by $V(G \times H) = V(G) \times V(H)$ and for $(a, b), (c, d) \in V(G \times H)$, $[(a, b), (c, d)] \in E(G \times H)$ if $a = c$ and $[b, d] \in E(H)$ or if $b = d$ and $[a, c] \in E(G)$.

The hypercube Q_n with 2^n vertices is defined recursively by $Q_1 = K_2$ and for $n \geq 2$, $Q_n = K_2 \times Q_{n-1}$. Since Q_n has a transitive automorphism group, $I(Q_n) = 0$. The hypercube Q_n offers a desirable configuration for parallel computation because processors must exchange messages in executing an algorithm, and the distance between any two vertices (representing processors) in the hypercube is at most n.

By contrast, an $m \times m$ mesh configuration (formed by taking the cartesian product of two isomorphic line graphs, each with m vertices) consists of m^2 vertices and has a maximum distance of $2m$. A $2^{\frac{n}{2}} \times 2^{\frac{n}{2}}$ mesh for even n having the same number of vertices as Q_n has a maximum distance between vertices of $2(2^{\frac{n}{2}} - 1)$. At the same time the information content of such a mesh is $\frac{n}{2} - 1$ [13].

This example suggests that good graph configurations for parallel computation score low on information complexity or, alternatively, are highly symmetric. Information complexity is a coarse filter, but it is useful nonetheless.

Computing the group-based entropy or information content of a graph requires knowledge of the orbits of the automorphism group. An obvious approach to computing the orbits is to determine the automorphism group and then to observe the action of automorphisms on the vertices of the graph. This is not an efficient method in general, but the algebraic structure of a graph can be exploited to find the automorphism group efficiently in some cases. The general question of determining the automorphism group is taken up in Section 1.3; heuristics for finding the orbits of $Aut(G)$ are surveyed in Section 1.4.

1.3
Groups and Graph Spectra

Let $G = (V, E)$ be a graph with vertex set V of size n, edge set E of size m, and automorphism group $Aut(G)$. (See [3] for general coverage of algebraic aspects of graph theory and [12] for specific treatment of the automorphism group of a graph.) Since automorphisms are in effect relabelings of the vertices, they can be represented as permutation matrices. Let $A = A(G)$ be the adjacency matrix of G. Then a permutation matrix P is an automorphism of G if and only if $P^T AP = A$ or $PA = AP$.

Thus, one way to construct the automorphism group of a graph G is to solve the matrix equation $AX = XA$ for permutation matrices X. The Jordan canonical form of A as a matrix over the reals can be used to obtain the general solution X. Taking G to be undirected and thus A symmetric and letting $\tilde{A} = U^T AU$ be the Jordan form of A, we have $(U\tilde{A}U^T)X = X(U\tilde{A}U^T)$ or $\tilde{A}\tilde{X} = \tilde{X}\tilde{A}$, where $\tilde{X} = U^T XU$.

Thus the construction of $Aut(G)$ requires computing the orthogonal matrix U and finding all $\tilde{X}$ that commute with $\tilde{A}$. The matrix $\tilde{X}$ depends on the elementary divisors of A. With no additional information, this method of constructing the group is not too promising since it is necessary to find all those solutions that are permutation matrices.

In the special case where A has all distinct eigenvalues, $\tilde{X}$ has the form of a diagonal matrix with arbitrary parameters on the main diagonal. In this case, $X = U\tilde{X}U^T$. Clearly $U\tilde{X}U^T$ is symmetric, so if it is a permutation matrix, it must correspond to a product of disjoint transpositions. This means that every element of $Aut(G)$ has order 2 and the group is therefore abelian [12, 17]. The converse is not true since, for example, the graph G of Figure 1.3 has the characteristic polynomial $(x+1)^2(x^3-2x^2-5x+2)$.

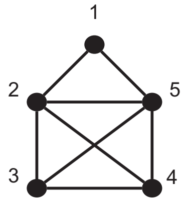


Figure 1.3 *Aut*(G) is abelian, every element is of order 2, but the characteristic polynomial has repeated roots.

An analogous result holds for digraphs. Using the same analysis, Chao [5] showed that if the adjacency matrix of a digraph has all distinct eigenvalues, then its automorphism group is abelian. However, the automorphisms need not be of order 2. For example, the adjacency matrix of digraph D in Figure 1.4 has the characteristic polynomial $(x^3-1) = (x-1)(x^2+x+1)$ but the permutation (123) is an automorphism of D.

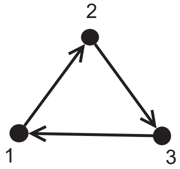


Figure 1.4 *Aut*(D) $= \langle(123)\rangle$, abelian but not every element has order 2.

Both of these results are special cases of the following:

Theorem 1.1 *Suppose the adjacency matrix $A = A(D)$ of a digraph D is non-derogatory with respect to a field F, i.e., its characteristic polynomial coincides with its minimal polynomial over F. Then Aut(D) is abelian.*

Proof. The result is an immediate consequence of the fact that under the hypothesis of the theorem, every matrix over F commuting with A can be expressed as a polynomial in A.

In particular, if A has all distinct eigenvalues, it is non-derogatory over the complex number field. To see that every automorphism of an (undirected) graph has order 2 under this condition, it suffices to observe that any polynomial in a symmetric matrix is again symmetric.

If the adjacency matrix fails to be nonderogatory, then some leverage in constructing the automorphism group can be obtained by taking advantage of the fact that the matrix consists of zeroes and ones. In particular, the adjacency matrix can be interpreted as a matrix over $GF(2)$, thus reducing the solution space of the matrix equation $AX = XA$ to zero-one matrices at the outset.

Thus suppose $A = A(G)$ (for a graph G) is a matrix over $GF(2)$. To demonstrate a method for constructing automorphisms, we revisit the special case of A being nonderogatory over $GF(2)$.

In this case we know that:

1. $M \in Aut(G)$ implies $M^2 = I$ (the identity matrix) and
2. $M \in Aut(G)$ implies $M = \sum_{i=0}^{n-1} a_i A^i$.

So if $M \in Aut(G)$, then we can write

$$M = \sum_{i=0}^{n-1} a_i A^i$$

and

$$I = M^2 = \left(\sum_{i=0}^{n-1} a_i A^i \right)^2 = \sum_{i=0}^{n-1} a_i (A^i)^2.$$

Thus $\{M | M = \sum_{i=0}^{n-1} a_i A^i \text{ and } M^2 = I\} \supseteq Aut(G)$.

Constructing the group in this case reduces to finding all polynomials in A^2 that are equal to the identity matrix. These have the form

$$p(A)\mu_{A^2}(A^2) + I,$$

where $\mu_{A^2}(x)$ is the minimal polynomial of A^2.

Thus, if $M^2 = I$, then $M = p(A)\mu_{A^2}(A) + I$ for some polynomial $p(x)$, since $(p(A)\mu_{A^2}(A) + I)^2 = (p(A^2)\mu_{A^2}(A^2) + I) = 0 + I = I$.

The characteristic and minimal polynomials of graph G in Figure 1.5 coincide over the real numbers, i.e., $\phi(x) = \mu(x) = (x^3 - x^2 - 6x + 2)x(x + 1)$ and over $GF(2)$ with $\phi(x) = \mu(x) = x^3(x + 1)^2$. Hence, the adjacency matrix of G is nonderogatory over both fields. The minimal polynomial of A^2 is $\mu_{A^2}(x) = x^2(x + 1)$, which is of degree 3.

Therefore, $M \in Aut(G)$ implies $M = \mu_{A^2}(A)(b_0 I + b_1 A) + I$. There are four possible solutions for M corresponding to the four possible values for b_0 and b_1. All of these solutions, namely,

$$I, A^3 + A^2 + I, A^4 + A^3 + I, A^4 + A^2 + I,$$

turn out to be permutation matrices so that the automorphism group of G contains precisely these four elements.

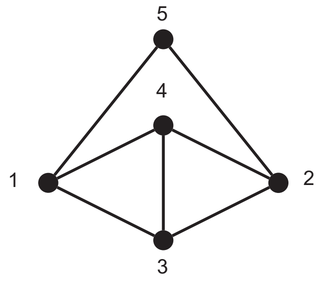


Figure 1.5 Computation of automorphisms over $GF(2)$.

Note that $\mu^2_{A^2}(x) = x\phi_A(x)$ if n is odd, or $\mu^2_{A^2}(x) = \phi_A(x)$ if n is even. Hence, if $m = \deg \mu_{A^2}(x)$ and M satisfies $AM = MA$ and $M^2 = I$, then $M = \mu_{A^2}(x) \sum_{i=0}^{n-m-1} b_i A^i + I$, where $b_i \in GF(2)$.

To determine $Aut(G)$, it suffices to examine $2^{n-m-1} \approx 2^{n/2}$ values of the parameters b_i, to pick out the permutation matrices (i.e., elements of $Aut(G)$).

However, some further simplification is possible. Let $Q = \mu_{A^2}(A)$ and $Z(b) = \sum_{i=0}^{n-m-1} b_i A^i$. Then $M = QZ(b) + I$. Multiplying by M on the right gives $MQ = Q^2 Z(b) + Q = Q$. Thus, if M is an automorphism of G, then $MQ = Q$, which means that similar vertices of G correspond to identical rows of Q. In addition, the identical rows must occur in *minimal pairs*, which gives a sufficient condition for $Aut(G)$ to be trivial.

If $\mu_{A^2}(A)$ has all distinct rows or no minimal pairs of identical rows, then $Aut(G)$ is trivial. The converse is not true. Both graphs in Figure 1.6 have trivial groups, but $\mu_{A^2}(A(G_1))$ has all distinct rows while $\mu_{A^2}(A(G_2))$ has three pairs of identical rows.

Theorem 1.2 *[18]; see also [6]. Let D be a digraph and $A = A(D)$ be its adjacency matrix. If $\phi_A(x)$ is irreducible over the integers, then $Aut(D)$ is trivial.*

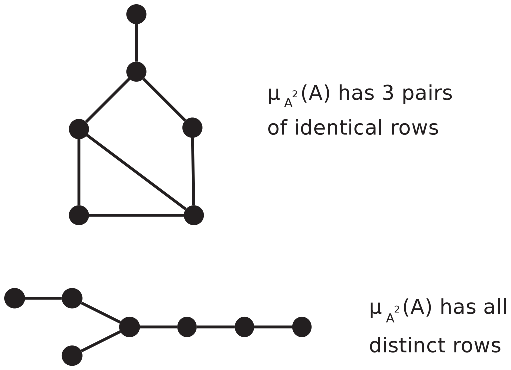

$\mu_{A^2}(A)$ has 3 pairs of identical rows
$\mu_{A^2}(A)$ has all distinct rows

Figure 1.6 Identity graphs.

Proof. Suppose there is an $M(\neq I) \in Aut(D)$, and that the permutation corresponding to M consists of r disjoint cycles of lengths $k_1, \ldots, k_r$. Let z be a nonzero vector consisting of k_1 components equal to c_1, followed by k_2 components equal to c_2, followed by $\ldots k_r$ components equal to c_r. Consider $Az = xz$. This gives a system of r equations in the r unknowns $c_1, c_2, \ldots, c_r$. Thus $Az = xz$ reduces to $Bc = xc$, where $c = (c_1, c_2, \ldots, c_r)^T$. Now z and c are eigenvectors of A and B, respectively, and $\det(B - xc) | \det(A - xz)$, where $\deg(\det(B - xc)) < \deg(\det(A - xz))$. Hence, $\phi_A(x)$ has a nontrivial factorization, which completes the proof.

Figure 1.7 shows a digraph (D) and graph (G) (with the smallest number of vertices) satisfying the condition of the theorem. $\phi_{A(D)}(x) = x^3 - x - 1$ and $\phi_{A(G)}(x) = x^6 - 6x^4 - 2x^3 + 7x^2 + 2x - 1$.

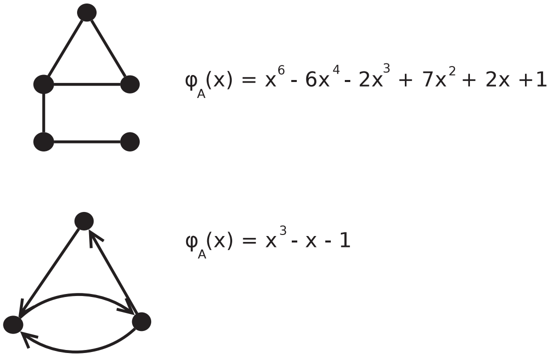

$\varphi_A(x) = x^6 - 6x^4 - 2x^3 + 7x^2 + 2x +1$
$\varphi_A(x) = x^3 - x - 1$

Figure 1.7 Smallest graph and digraph whose characteristic polynomials are irreducible over the integers.

Note that the theorem also holds if $\phi_A(x)$ is taken as a polynomial over a finite field. For example, over $GF(2)$, $x^3 - x - 1$ is irreducible, but $x^6 - 6x^4 - 2x^3 + 7x^2 + 2x - 1 = x^6 + x^2 + 1 = (x^3 + x + 1)^2$.

For graphs this criterion is not very useful since the characteristic polynomial of any graph is reducible over $GF(2)$. There are regular graphs and trees that have the trivial group, but the characteristic polynomial of any regular graph has a linear factor, as does the characteristic polynomial of a tree with an odd number of vertices.

The foregoing discussion suggests the utility of trying to relate the factorization of the characteristic polynomial to the structure of the automorphism group. For example, if G is a graph with an even number n of vertices and adjacency matrix $A = A(G)$, and if $\phi_A(x) = \alpha(x)\beta(x)$ with $\deg\alpha = \deg\beta$ and both α and β are irreducible, then either $Aut(G)$ is trivial or it is of order 2 and consists of the identity and (with a suitable labeling) the permutation $(1, 2)(3, 4) \cdots (n/2, n/2)$.

Table 1.1 contains a list of all 156 graphs on six vertices, showing factored characteristic polynomials and the sizes of their respective automorphism group orbits. Each graph is defined by its list of edges, shown as a sequence of pairs of numbers referring to a standard template with the vertices numbered from 1 to 6 in clockwise order. The last column shows the sizes of the orbits. Complements are not given explicitly, but their polynomials are listed. The orbits of G and $\bar{G}$ are the same.

Table 1.1 Characteristic polynomials and orbit sizes of all graphs on six vertices.

G: # edges: list	Polynomial of G	Polynomial of $\bar{G}$	Orbit sizes
0:	x^6	$(x+1)^5(x-5)$	6
3: 16 23 45	$(x+1)^3(x-1)^3$	$x^3(x+2)^2(x-4)$	6
6: 12 16 23 34 45 56	$(x-1)^2(x+1)^2(x+2)(x-2)$	$x^2(x-1)(x+2)^2(x-3)$	6
6: 15 16 23 24 34 56	$(x-2)^2(x+1)^4$	$x^4(x-3)(x+3)$	6
3: 15 16 56	$x^3(x+1)^2(x-2)$	$x^2(x+1)^2(x^2-2x-9)$	33
1: 12	$x^4(x-1)(x+1)$	$x(x+1)^3(x^2-3x-8)$	24
4: 12 15 24 45	$x^4(x-2)(x+2)$	$(x-1)(x+1)^3(x^2-2x-7)$	24
4: 12 16 34 45	$x^2(x^2-2)^2$	$(x+1)^2(x^4-2x^3-8x^2+6x-1)$	24
5: 12 15 16 23 24	$x^2(x-1)(x+1)(x-2)(x+2)$	$(x-1)(x+2)(x+1)^2(x^2-3x-2)$	24
5: 14 16 23 45 56	$x^2(x-1)(x+1)(x-2)(x+2)$	$x(x-1)(x+1)^2(x^2-x-8)$	24
6: 12 14 15 24 25 45	$x^2(x-3)(x+1)^3$	$x^3(x+1)(x^2-x-8)$	24
7: 12 16 23 25 34 45 56	$(x+1)(x-1)(x^2-2x-1)(x^2+2x-1)$	$x(x+2)(x^2-2)(x^2-2x-2)$	24
7: 14 15 16 23 45 46 56	$(x+1)^4(x-3)(x-1)$	$x^4(x^2-8)$	24
7: 15 16 23 24 34 45 56	$(x+1)^2(x^2-3)(x^2-2x-1)$	$x^2(x^2-2x-2)(x^2+2x-2)$	24
2: 12 56	$x^2(x-1)^2(x+1)^2$	$x^2(x+1)(x+2)(x^2-3x-6)$	222
3: 12 16 23	$x^2(x^2-x-1)(x^2+x-1)$	$(x+1)(x^2+x-1)(x^3-2x^2-8x-3)$	222
6: 14 15 16 23 45 56	$x(x-1)(x+1)^2(x^2-x-4)$	$x^2(x+1)(x^3-x^2-8x+4)$	222
7: 13 16 23 26 34 45 56	$x(x^2+x-1)(x^3-x^2-5x+4)$	$(x-1)(x+1)(x^2+x-1)(x^2-x-5)$	222
7: 12 15 23 24 25 45 56	$x^2(x^2+x-1)(x^2-x-5)$	$(x+1)(x^2+x-1)(x^3-2x^2-4x+1)$	222
5: 12 16 24 45 56	$x(x-2)(x^2+x-1)^2$	$(x^2+x-1)^2(x^2-2x-5)$	15
5: 12 23 24 25 26	$x^4(x^2-5)$	$x(x-4)(x+1)^4$	15
2: 12 16	$x^4(x^2-1)$	$(x+1)^3(x^3-3x^2-7x+3)$	123
3: 12 15 16	$x^4(x^2-3)$	$(x+1)^3(x^3-3x^2-6x+4)$	123
4: 15 16 23 56	$x(x+1)(x-2)(x+1)^3$	$x^3(x^3-11x-12)$	123
4: 12 15 16 34	$x^2(x+1)(x-1)(x^2-3)$	$x(x+1)^2(x^3-2x^2-8x+4)$	123

Table 1.1 (continued).

G: # edges: list	Polynomial of G	Polynomial of $\bar{G}$	Orbit sizes
5: 15 16 23 34 56	$x(x-2)(x+1)^2(x^2-2)$	$x^2(x+1)(x^3-x^2-9x+3)$	123
6: 12 15 23 24 35 45	$x^4(x^2-6)$	$(x+1)^3(x^3-3x^2-3x+7)$	123
6: 12 13 14 15 16 56	$x^2(x+1)(x^3-x^2-5x+3)$	$x^2(x+1)^2(x^2-2x-6)$	123
7: 12 14 15 16 25 45 56	$x^3(x-3)(x+1)(x+2)$	$x(x-1)^2(x^3-2x^2-5x+4)$	123
4: 12 14 15 16	$x^4(x-2)(x+2)$	$(x+1)^3(x^3-3x^2-5x+3)$	114
7: 15 16 26 34 35 45 56	$(x-1)(x+1)^2(x^3-x^2-5x+1)$	$x^3(x+2)(x^2-2x-4)$	114
3: 16 23 56	$x^2(x+1)(x-1)(x^2-2)$	$x(x+1)(x^4-x^3-11x^2-7x+4)$	1122
4: 12 15 16 56	$x^2(x+1)(x^3-x^2-3x+1)$	$x(x+1)(x+2)(x^3-3x^2-4x+2)$	1122
4: 12 16 23 56	$x^2(x-1)(x+1)(x^2-3)$	$x(x+1)(x+2)(x^3-3x^2-4x+4)$	1122
5: 12 15 16 45 56	$x^2(x^2-x-3)(x^2+x-1)$	$(x+1)(x^2+x-1)(x^3-2x^2-6x+1)$	1122
5: 12 14 15 16 56	$x^2(x+1)(x^3-x^2-4x+2)$	$x(x+1)(x^4-x^3-9x^2-5x+4)$	1122
5: 12 15 16 23 45	$(x-1)(x+1)(x^4-4x^2+1)$	$x(x+2)(x^4-2x^3-6x^2+2x+4)$	1122
5: 15 16 23 45 56	$(x-1)(x+1)^2(x^3-x^2-3x+1)$	$x^2(x^4-1x^2-8x+4)$	1122
6: 12 15 16 24 45 56	$x^2(x+2)(x^3-2x^2-2x+2)$	$(x-1)(x+1)(x^4-8x^2-8x+1)$	1122
6: 13 16 23 34 45 56	$(x-1)(x^2+x-1)(x^3-4x-1)$	$(x+2)(x^2+x-1)(x^3-3x^2-x+2)$	1122
6: 12 13 14 16 45 56	$x^2(x^4-6x^2+4)$	$(x-1)(x+1)^2(x^3-x^2-7x-3)$	1122
4: 12 16 23 45	$(x-1)(x+1)(x^2-x-1)(x^2+x-1)$	$x(x^2+x-1)(x^3-x^2-9x-4)$	222
5: 12 14 15 24 45	$x^3(x+1)(x^2-x-4)$	$x(x+1)^2(x^3-2x^2-7x+4)$	222
5: 12 16 34 45 56	$(x^3-x^2-2x+1)(x^3+x^2-2x-1)$	$(x^3-2x^2-5x+1)(x^3+2x^2-x-1)$	222
6: 12 14 16 34 45 56	$(x^3-2x^2-x+1)(x^3+2x^2-x-1)$	$(x^3-x^2-6x-3)(x^3+x^2-2x-1)$	222
6: 12 15 16 23 25 45	$(x^2-2x-1)(x^2+x-1)^2$	$(x^2-2x-4)(x^2+x-1)^2$	1122
6: 12 15 16 23 24 56	$x(x+1)(x^4-x^3-5x^2+3x+4)$	$x(x+1)(x^4-x^3-8x^2-2x+6)$	1122
7: 12 15 16 24 26 45 56	$x^2(x+1)(x^3-x^2-6x+2)$	$x(x+1)(x^4-x^3-7x^2+x+8)$	1122
7: 12 14 15 16 24 45 56	$x(x^2+x-1)(x^3-x^2-5x-2)$	$(x^2+x-1)(x^4-x^3-6x^2-x+1)$	1122
7: 12 16 23 24 34 45 56	$(x^2+x-1)(x^4-x^3-5x^2+2x+4)$	$(x^2+x-1)(x^4-x^3-6x^2+3x+1)$	1122
7: 14 16 23 24 34 45 56	$x(x+1)(x^4-x^3-6x^2+4x+4)$	$x(x-1)(x+1)(x^3-7x-4)$	1122
7: 12 13 15 24 34 45 56	$x^2(x^4-7x^2+4)$	$(x+1)^2(x^4-2x^3-5x^2+6x+4)$	1122
7: 12 14 16 23 24 45 56	$x^2(x-1)(x+2)(x^2-x-4)$	$(x+1)(x-1)(x+2)(x^3-2x^2-3x+2)$	1122

Table 1.1 (continued).

G:	Polynomial	Polynomial	Orbit
# edges: list	of G	of $\bar{G}$	sizes
7: 14 16 24 34 45 46 56	$x^2(x+2)(x^3-2x^2-3x+2)$	$x(x+1)^2(x^3-2x^2-5x+2)$	1122
7: 12 15 16 24 25 34 45	$(x-1)(x+1)^2(x^3-x^2-5x+1)$	$x^2(x+2)(x^3-2x^2-4x+4)$	1122
7: 12 15 16 23 24 25 56	$x(x-1)(x+1)(x+2)(x^2-2x-2)$	$x(x+1)(x^4-x^3-7x^2+x+4)$	1122
5: 16 25 35 45 56	$x^2(x^4-5x^2+3)$	$(x+1)^2(x^4-2x^3-7x^2+2x+3)$	1113
7: 12 14 15 24 25 34 45	$x(x+1)^2(x^3-2x^2-4x+2)$	$x^2(x^4-8x^2-6x+3)$	1113
7: 12 13 15 16 24 34 45	$x^2(x^4-7x^2+3)$	$(x+1)^2(x^4-2x^3-5x^2+4x+3)$	1113
4: 12 15 16 23	$x^2(x^4-4x^2+2)$	$(x+1)(x^5-x^4-1x^3-6x^2+7x+3)$	11112
5: 12 15 16 23 56	$x(x-1)(x+1)(x^3-4x-2)$	$x(x^5-1x^3-1x^2+5x+4)$	11112
5: 12 15 16 34 45	$x^2(x^4-5x^2+2)$	$(x+1)(x^5-x^4-9x^3-3x^2+1x+4)$	11112
5: 16 24 34 45 56	$x^2(x^4-5x^2+5)$	$(x+1)(x^5-x^4-9x^3-x^2+7x-1)$	11112
6: 12 15 23 24 25 45	$x^2(x^4-6x^2-4x+2)$	$(x+1)(x^5-x^4-8x^3-2x^2+5x-1)$	11112
6: 12 15 24 25 34 45	$x(x+1)(x^4-x^3-5x^2+x+2)$	$x(x+2)(x^4-2x^3-5x^2+2x+2)$	11112
6: 14 16 23 34 45 56	$x^2(x^4-6x^2+6)$	$(x+1)(x^5-x^4-8x^3+2x^2+9x-1)$	11112
6: 12 13 15 16 45 56	$x^2(x^4-6x^2-2x+5)$	$x(x+1)(x^4-x^3-8x^2-2x+6)$	11112
7: 12 14 15 16 23 45 56	$x(x-1)(x+1)(x+2)(x^2-2x-2)$	$(x+1)(x^5-x^4-7x^3+3x^2+3x-1)$	11112
7: 14 15 16 23 24 45 56	$x(x+1)^2(x^3-2x^2-4x+6)$	$x^2(x^4-8x^2-2x+7)$	11112
7: 15 16 24 25 34 45 56	$(x+1)(x^5-x^4-6x^3+2x^2+7x-1)$	$x^2(x^4-8x^2-4x+6)$	11112
7: 12 15 16 23 24 25 45	$x^6-7x^4-4x^3+6x^2+2x-1$	$x^6-8x^4-6x^3+7x^2+4x-1$	111111
7: 12 15 16 24 34 45 56	$x^6-7x^4-2x^3+8x^2+2x-1$	$x^6-8x^4-4x^3+9x^2+4x-1$	111111
7: 12 14 16 24 34 45 56	$x^6-7x^4-2x^3+7x^2-1$	$x(x^5-8x^3-6x^2+8x+6)$	111111

1.4 Approximating Orbits

The automorphism group $Aut(G)$ of a graph G is a subgroup of S_n, the symmetric group on n objects, so $|Aut(G)| \leq n!$. Constructing all the elements of the automorphism group could take exponential time, e.g., K_n has S_n as its automorphism group. However, it may be sufficient to find a relatively small generating set that represents $Aut(G)$. Indeed, it is always possible to find a generating set of size $\log n$ for a group H of size n [1].

Unfortunately it is not known whether or not such a small set representing $Aut(G)$ can be computed in polynomial time, because the problem of determining the automorphism group can be shown to be equivalent to graph isomorphism (i.e., determining whether two graphs are isomorphic). The relationship between the two problems is shown more explicitly in [1].

Since the problem of determining when two graphs are isomorphic has been studied extensively and is not known to be solvable by a polynomial bounded algorithm, heuristics are needed to find the orbits of the automorphism group. If such heuristics are easy to compute and provide a high degree of accuracy, the complexity of a graph can be computed efficiently with a high degree of confidence.

The orbits of a graph consist of vertices with similar properties such as having the same degree. So if it were possible to create a small list of all these properties and if, in addition, there were polynomial time tests for each one, then there would be a polynomial time algorithm for the graph automorphism problem. Of course such a complete list of properties is not known. However, if there exists one such property that does not hold for two vertices, then these vertices are not in the same orbit. So, creating a partial list of polynomial time tests would help to distinguish vertices having different properties and thus to separate them into different orbits. In surveying the literature on heuristic approaches to computing the orbits of the automorphism group of a graph we have made use of [20], which in turn draws on [7].

The procedure adopted here for finding the orbits of a graph is as follows:

1. Identify several polynomially checkable properties designed to distinguish between vertices. At the start of the procedure all the vertices are taken to be in the same orbit.
2. For each property and each pair of vertices u, v in an orbit thus far determined, find whether or not u and v can be distinguished by the property. If yes, then draw the inference "u, v are in different orbits"; otherwise, apply the next test.

If two vertices pass all the tests, then they will be considered to be in the same orbit. This procedure gives rise to a deterministic process with one-sided error, i.e., two vertices in the same putative orbit may in fact be distinguishable.

Critical to developing an efficient procedure is making judicious choices of vertex properties that can serve as tests. The selection of properties used in our procedure has been guided by results in the theory of networks and in sociological theory.

1.4.1 The Degree of the Vertices

The first test is quite simple: If two vertices have different degrees, then they cannot be in the same orbit. The degree of a vertex can be computed in time n, with n being the size of the graph. So this test will take time $O(n)$. The degree of a vertex is an important property in the theory of networks since finding the high-degree vertices in an underlying graph is considered equivalent to determining the so-called "authorities" of the network.

1.4.2 The Point-Deleted Neighborhood Degree Vector

Examining only the degree of the vertices is insufficient. Consider, for example, the path of five vertices, labeled 1, 2, 3, 4, 5. The degree test does not distinguish 2 and 4 from 3, but 3 is in not in the same orbit as 2 and 4; this is obvious since both 2 and 4 have one neighbor (1 and 5, respectively) with degree 1, while 3 does not have such a neighbor. This observation leads naturally to the idea of the second test, namely, to examine the neighborhood of the vertices.

Definition 1.1 The *neighborhood* of a subset $S \subseteq V$ denoted by $N(S)$ is the set of all the neighbors of S, i.e., $N(S) = S \bigcup \{v \in V \setminus S | (v, s) \in E \text{ and } s \in S\}$. The $\{i + 1\}^{th}$ *degree neighborhood* of S is defined inductively, $N^{i+1}(S) = N(N^i(S))$. The *point-deleted neighborhood* of v, with v being a vertex of V, is $\tilde{N}^i(x) = N^i(\{x\}) \setminus \{x\}$.

Thus, for the path mentioned above we have that

- $N(\{3\}) = \{2, 3, 4\}$,
- $N^2(\{3\}) = \{1, 2, 3, 4, 5\}$,
- $\tilde{N}(3) = \{2, 4\}$.

The notion of the degree vector is also needed.

Definition 1.2 The *degree vector* $d(S)$ of a subset S of the vertices of a graph can be defined as the ordered sequence of the degrees of these vertices in the induced subgraph.

Again for the above example of the path of five vertices, the degree vector of the whole path is (1,1,2,2,2). The degree vector of {2,3} is (1,1).

The test to be considered is comparing the point-deleted neighborhood degree vectors of the vertices to be examined. This technique is presented in [7]. The degree test yields several groups of vertices of equal degree. To apply this test we first compute the point-deleted neighborhood of each of the vertices to be compared and then determine the degree vector. If the degree vector is different, draw the inference "not in the same orbit." The test can be made more subtle if higher-order neighborhoods are taken into account.

The execution time required for this test can be computed as follows. If v and u are the vertices to be compared and k is their degree, we can compute the degree vectors of the point-deleted neighborhoods of u and v in time k^2. Sorting the two vectors, each of size k, and comparing them is at most of this order of complexity. Thus the total execution time required is $\Theta(k^2)$. Since k is bounded by n, the worst-case complexity of the test is $O(n^2)$.

As an example, consider the graph of Figure 1.8. Table 1.2 shows the point-deleted neighborhood of the vertices.

The test defines two groups of vertices that for this example coincide with the two orbits. The sets are {1,3,5,7} and {2,4,6,8}.

Examining the point-deleted neighborhood degree vector has several advantages and disadvantages over the degree sequence of the neighbors of a vertex. The latter would probably give faster negative results when testing whether two vertices belong in the same orbit, since it takes account of the whole graph and not simply the induced subgraph of the neighbors; on the other hand, the former also works in regular graphs. However, the main reason for using the latter test is that it is associated with the concept of clustering coefficients that are widely used in networking theory. Informally

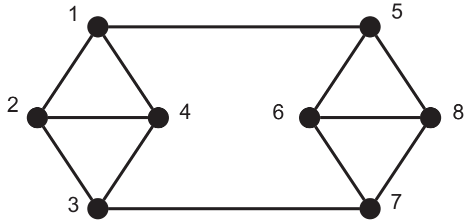


Figure 1.8 Example: graph for point-deleted neighborhood degree vectors test.

Table 1.2 Information about the point-deleted neighborhood of every vertex and its degree vector.

v	$\tilde{N}(v)$	$d(\tilde{N}(v))$
1	{2,4,5}	(0,1,1)
2	{1,3,4}	(1,1,2)
3	{2,4,7}	(0,1,1)
4	{1,2,3}	(1,1,2)
5	{1,6,8}	(0,1,1)
6	{5,7,8}	(1,1,2)
7	{3,6,8}	(0,1,1)
8	{5,6,7}	(1,1,2)

speaking, the clustering coefficient indicates the degree to which the induced subgraph of the neighbors of a vertex resembles a clique. Determining the point-deleted neighborhood degree vector of a vertex would help in computing the clustering coefficient. So vertices that pass this test will also have the same clustering coefficient.

1.4.3 Betweenness Centrality

The above techniques correctly determine the orbits of a large variety of graphs. However, there are cases where they fail. Consider, for example, the graph in Figure 1.9. The point-deleted neighborhood degree vector would place vertices 13 and 14 in the same orbit since $d(\tilde{N}(13)) = d(\tilde{N}(14)) = (2, 2, 2, 2, 3, 3)$. However, it is obvious from the figure that vertices 13 and 14 do not belong in the same orbit.

In this section we will describe one more method of estimating the orbits of a graph. This method is based on the concept of betweenness centrality, which was first introduced in [8] and can be described as follows.

Definition 1.3 The betweenness centrality of a vertex x of a graph $G(V, E)$ is the sum over all pairs of vertices y, z in the graph of the number of shortest paths ($p^x_{y,z}$) from y to z that pass through x divided by the number of all

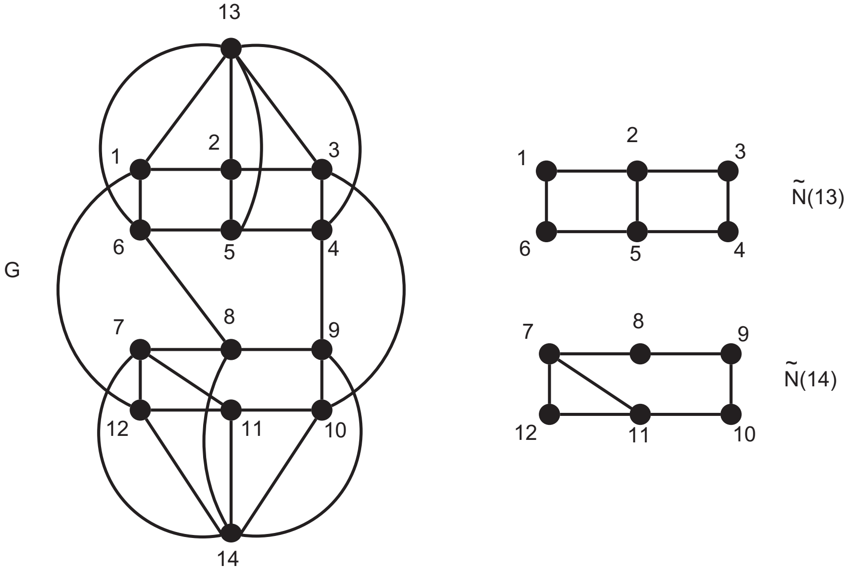


Figure 1.9 Graph where the technique of the point-deleted neighborhood degree vector fails.

the shortest paths from y to z. More precisely, the betweenness centrality of a vertex x is:

$$C_B(x) = \frac{1}{2} \sum_{y \in V} d_{y,x} ,$$

where

$$d_{y,x} = \sum_{z \in V} b_{y,z}$$

and

$$b_{y,z}(x) = \frac{p^x_{y,z}}{p_{y,z}} .$$

This measure is suggested by the sociology of networks of individuals, so-called “social networks.” In addition, it captures important structural features of a graph, making it quite useful in approximating orbits.

The method considered in [7] is an extension of the point-deleted neighborhood degree vector. Once the induced subgraph on $\tilde{N}(x)$ is computed for every vertex x, we compute the betweenness vector $C_B(\tilde{N}(x)) = C_B(v_1)$, $C_B(v_2), \ldots, C_B(v_{d(x)})$, where $C_B(v_i) \leq C_B(v_{i+1})$, with $v_i \in \tilde{N}(x)$ and $d(x)$ being the degree of vertex x. If the vectors $C_B(\tilde{N}(x))$ and $C_B(\tilde{N}(y))$ are not identical, then x and y belong to different orbits.

Consider again the graph of Figure 1.9. We want to compute the betweenness centrality vectors of vertices 13 and 14, which will help in deciding whether they belong in the same orbit or not. We first find the induced subgraphs on $\tilde{N}(x)$ for every vertex x (also shown in Figure 1.9). Then, for each subgraph, we first compute a table that contains all the intermediate vertices in every shortest path between each pair of vertices. We assume that paths of length 1 have no intermediate vertices and we omit paths that start and end on the same vertex. The tables are shown below (Table 1.3). Then we compute the dependency matrix, the matrix $\boldsymbol{D} = d_{y,z}$ for every y, z. Finally we sum and then halve every column of $\boldsymbol{D}$ to compute the betweenness centrality of every vertex of the induced subgraph. The dependency matrices for both vertices 13 and 14 are computed below (Table 1.4). The computed vector will be the betweenness centrality vector of vertex x. Finally, when all the computations are done, we compute the vectors of each pair of vertices. If they are different, then the vertices belong in different orbits; otherwise the algorithm concludes that the vertices belong to the same orbit.

It is clear that

$$C_B(\tilde{N}(13)) = \left(\frac{5}{6}, \frac{5}{6}, \frac{5}{6}, \frac{5}{6}, 3, 3\right) \neq C_B(\tilde{N}(14)) = \left(0, 1, 1\frac{1}{2}, 1\frac{1}{2}, 2\frac{1}{2}, 2\frac{1}{2}\right) .$$

Thus vertices 13 and 14 belong to different orbits.

Table 1.3 Tables containing the intermediate vertices of the shortest paths between each pair of vertices.

	1	2	3	4	5	6
1	–	–	2	2 3 2 5 6 5	2 6	–
2	–	–	–	3 5	–	1 5
3	2	–	–	–	2 4	2 1 2 5 4 5
4	3 2 5 2 5 6	3 5	–	–	–	5
5	2 6	–	2 4	–	–	–
6	–	1 5	1 2 5 2 5 4	5	–	–

	7	8	9	10	11	12
7	–	–	8	11	–	–
8	–	–	–	9	7	7
9	8	–	–	–	10	8 7 10 11
10	11	9	–	–	–	11
11	–	7	10	–	–	–
12	–	7	8 7 10 11	11	–	–

Table 1.4 Dependency matrices for vertices 13 and 14 and the betweenness centrality vectors.

	1	2	3	4	5	6
1	0	$1\frac{5}{6}$	$\frac{1}{3}$	0	$\frac{2}{3}$	$\frac{5}{6}$
2	$\frac{1}{2}$	0	$\frac{1}{2}$	0	1	0
3	$\frac{1}{3}$	$1\frac{5}{6}$	0	$\frac{5}{6}$	$\frac{2}{3}$	0
4	0	$\frac{2}{3}$	$\frac{5}{6}$	0	$1\frac{5}{6}$	$\frac{1}{3}$
5	0	1	0	$\frac{1}{2}$	0	$\frac{1}{2}$
6	$\frac{5}{6}$	$\frac{2}{3}$	0	$\frac{1}{3}$	$1\frac{5}{6}$	0
C_B	$\frac{5}{6}$	3	$\frac{5}{6}$	$\frac{5}{6}$	3	$\frac{5}{6}$

	7	8	9	10	11	12
7	0	1	0	0	1	0
8	2	0	1	0	0	0
9	$\frac{1}{2}$	$1\frac{1}{2}$	0	$1\frac{1}{2}$	$\frac{1}{2}$	0
10	0	0	1	0	2	0
11	1	0	0	1	0	0
12	$1\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	0
C_B	$2\frac{1}{2}$	$1\frac{1}{2}$	1	$1\frac{1}{2}$	$2\frac{1}{2}$	0

The betweenness centrality of all the vertices of a graph can be computed in $O(n^3)$ time (where n is the number of vertices in the graph) by a modified version of Floyd's algorithm for determining all shortest paths between pairs of vertices. The fastest known exact algorithm for determining the betweenness centrality of all the vertices of a graph is due to Brandes [4], and its complexity is $\Theta(n \cdot m)$, where m is the number of edges of the graph. Thus, if u and v are vertices to be compared and k is their degree, then determining the betweenness centrality vectors of u and v requires $O(k^3)$ time. Thus the worst-case complexity of the test is $O(n^3)$.

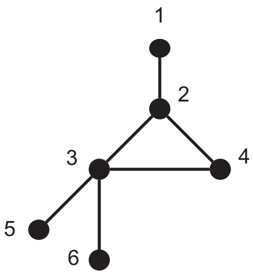
It is still possible to compute the orbits of the previous example exactly by examining the degree vector of higher-order point-deleted neighborhoods.

However, there are (rare) cases where the idea of the degree vector does not work at all. The graph (whose adjacency matrix is presented below) is such an example.

$$\begin{array}{cccccccccccccccccccccccccc}
0&1&0&1&1&0&0&0&0&1&1&0&1&1&0&0&0&1&0&0&0&0&0&1&0&1\\
1&0&1&0&1&1&0&0&0&0&1&1&0&1&1&0&0&0&1&0&0&0&0&0&1&0\\
0&1&0&1&0&1&1&0&0&0&0&1&1&0&1&1&0&0&0&1&0&0&0&0&0&1\\
1&0&1&0&1&0&1&1&0&0&0&0&1&1&0&1&1&0&0&0&1&0&0&0&0&0\\
1&1&0&1&0&1&0&1&1&0&0&0&0&0&1&0&1&1&0&0&0&1&0&0&0&0\\
0&1&1&0&1&0&1&0&1&1&0&0&0&0&0&1&0&1&1&0&0&0&1&0&0&0\\
0&0&1&1&0&1&0&1&0&1&1&0&0&0&0&0&1&0&1&1&0&0&0&1&0&0\\
0&0&0&1&1&0&1&0&1&0&1&1&0&0&0&0&0&1&0&1&1&0&0&0&1&0\\
0&0&0&0&1&1&0&1&0&1&0&1&1&0&0&0&0&0&1&0&1&1&0&0&0&1\\
1&0&0&0&0&1&1&0&1&0&1&0&1&1&0&0&0&0&0&1&0&1&1&0&0&0\\
1&1&0&0&0&0&1&1&0&1&0&1&0&0&1&0&0&0&0&0&1&0&1&1&0&0\\
0&1&1&0&0&0&0&1&1&0&1&0&1&0&0&1&0&0&0&0&0&1&0&1&1&0\\
1&0&1&1&0&0&0&0&1&1&0&1&0&0&0&0&1&0&0&0&0&0&1&0&1&1\\
1&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&1&1&1&0&0&1&0\\
0&1&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&1&1&1&0&0&1\\
0&0&1&1&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&1&1&1&1&0&0\\
0&0&0&1&1&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&1&1&1&1&0\\
1&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&1&1&1&1\\
0&1&0&0&0&1&1&0&1&0&0&0&0&1&0&0&1&0&0&0&1&0&0&1&1&1\\
0&0&1&0&0&0&1&1&0&1&0&0&0&1&1&0&0&1&0&0&0&1&0&0&1&1\\
0&0&0&1&0&0&0&1&1&0&1&0&0&1&1&1&0&0&1&0&0&0&1&0&0&1\\
0&0&0&0&1&0&0&0&1&1&0&1&0&1&1&1&1&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&1&0&0&0&1&1&0&1&0&1&1&1&1&0&0&1&0&0&0&1&0\\
1&0&0&0&0&0&1&0&0&0&1&1&0&0&0&1&1&1&1&0&0&1&0&0&0&1\\
0&1&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1&1&1&1&0&0&1&0&0&0\\
1&0&1&0&0&0&0&0&1&0&0&0&1&0&1&0&0&1&1&1&1&0&0&1&0&0
\end{array}$$

This graph is 10-regular, so that for every vertex x, $\tilde{N}(x)$ contains ten vertices. Furthermore, its diameter is two, so $\tilde{N}^2$ contains every vertex of the graph except for x. It follows that $d(\tilde{N}(x)) = (3, 3, 3, 3, 3, 3, 3, 3, 3, 3)$ for every vertex x. Thus the above tests will yield one orbit $\{1, \ldots, 26\}$. However, it can be proven that the graph has two orbits, $\{1, \ldots, 13\}$ and $\{14, \ldots, 26\}$ (this example is taken from [7]). The problem is solved by examining the point-deleted neighborhood betweenness centrality vector. We can show that $C_B(\tilde{N}(1)) = \ldots = C_B(\tilde{N}(13)) = (3, 3, 3, 3, 4, 4, 4, 5, 5, 5)$, whereas $C_B(\tilde{N}(14)) = \ldots = C_B(\tilde{N}(26)) = (3, 3, 3, 4, 4, 4, 4, 4, 4, 6)$.

In this section we have only examined the betweenness centrality measure in approximating the orbits of a graph. However, there are several variants of centrality measures that could be taken into account. These variants include



Chromatic decompositions:
{1,3} {2,5} {4,6}
{2} {1,3} {4,5,6}
{2} {3} {1,4,5,6}

Figure 1.10 Graph with multiple chromatic decompositions.

closeness centrality [22], graph centrality [9], and stress centrality [23]. All these concepts are attempts to capture the notion of the relative importance of a vertex in the overall structure of a graph, and thus each of them could play an important role in estimating the orbits of a graph.

1.5 Alternative Bases for Structural Complexity

Colorings of a graph can be used to obtain a decomposition of the vertices. Sets of vertices of the same color (or independent sets) constitute equivalence classes. Unlike the orbits of the automorphism group, a partition of the vertices obtained in this way is not unique. However, an information measure may be defined by taking the minimum value over some set of decompositions linked to colorings [16]. This section explores such a measure, compares it with the symmetry-based measure, and shows its relationship to the graph entropy as defined in [11].

A *coloring* of a graph is an assignment of colors to the vertices so that no two adjacent vertices have the same color.

An *n-coloring* of a graph $G = (V, E)$ is a coloring with n colors or, more precisely, a mapping f of V onto the set $\{1, 2, \ldots, n\}$ such that whenever $[u, v] \in E, f(u) \neq f(v)$.

The *chromatic number* $\kappa(G)$ of a graph G is the smallest value of n for which there is an n-coloring. A graph may have more than one n-coloring.

An n-coloring is *complete* if, for every i, j with $i \neq j$, there exist adjacent vertices u and v such that $f(u) = i$ and $f(v) = j$.

A decomposition $\{V_i\}_{i=1}^{n}$ of the set of vertices V is called a *chromatic decomposition* of G if $u, v \in V_i$ imply that $[u, v] \notin E$. Note that V_i in a chromatic decomposition is a set of *independent* vertices. If f is an n-coloring, the collection of sets $\{v \in V | f(v) = i\}_{i=1}^{n}$ forms a chromatic decomposition; conversely, a chromatic decomposition $\{V_i\}_{i=1}^{n}$ determines an n-coloring f. The sets V_i are thus called *color classes*.

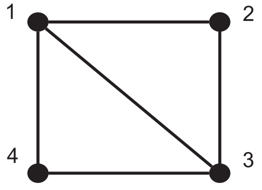
Given a graph $G = (V, E)$ with $|V| = n$ and $h = \kappa(G)$, let $\hat{V} = \{V_i\}_{i=1}^{h}$ be an arbitrary chromatic decomposition of G with $n_i(\hat{V}) = |V_i|$ for $1 \leq i \leq h$. The *chromatic information content* $I_c(G)$ of G is defined by the following formula [16]:

$$I_c(G) = \min_{\hat{V}} \left\{ -\sum_{i=1}^{h} \frac{n_i(\hat{V})}{n} \log \frac{n_i(\hat{V})}{n} \right\}.$$

Figure 1.10 shows a graph with three different chromatic decompositions whose finite probability schemes are $(1/3, 1/3, 1/3)$, $(1/2, 1/3, 1/6)$, and $(2/3, 1/6, 1/6)$. The minimum entropy is given by $(2/3, 1/6, 1/6)$, so that $I_c(G) = 2/3 \log 3/2 + 1/3 \log 6$.

$I_c(G)$ is defined as the minimum value over chromatic decompositions with $\kappa(G)$ color classes and thus does not necessarily give the minimum over all chromatic decompositions. When the graph does not have a complete k-coloring for $k > \kappa(G)$, $I_c(G)$ does give the minimum over all chromatic decompositions [16]. The restricted minimization in the definition allows for interpreting $I_c(G)$ as the amount of information needed to construct a $\kappa(G)$-coloring.

A related measure called *graph entropy* was introduced in [11] and subsequently applied to a variety of problems in graph theory and combinatorics [24]. This measure is a generalization of $I_c(G)$ formulated as average mutual information between two random variables representing the vertices and independent sets of G, respectively. Let S be the collection of independent sets of $G = (V, E)$ with $|V| = n$, and let $\boldsymbol{P}$ be a probability distribution on V. The graph entropy $H(G, \boldsymbol{P})$ is given by $I(V; S)$, the average mutual information between V and S (treated as random variables). Now, $I(V; S) = H(V) - H(V|S)$, so if $\boldsymbol{P}$ is a uniform probability distribution over V, then $H(G, \boldsymbol{P}) = I(V; S) = \log n - H(S)$. So, $I_c(G) = \log n - H(G, \boldsymbol{P})$. In summary, the essential difference between the two measures is that $I_c(G)$ as-



$$S = \{\{2,4\}, \{1\}, \{3\}\}$$

$$V = \{1,2,3,4\}$$

$$Pr[\{2,4\}] = \frac{1}{2}$$

$$Pr[\{1\}] = Pr[\{3\}] = \frac{1}{4}$$

$$Pr[v|s_k] = \begin{cases} 0, & \text{if } v \text{ is not in } s_k; \\ \frac{1}{|s_k|}, & \text{if } v \text{ is in } s_k. \end{cases}$$

$$H(G, P) = \log 4 - \left(\frac{1}{2} \log 2 + \frac{1}{2} \log 4 \right) = \frac{1}{2}$$

$$I_c(G) = \log 4 - H(G, P) = \frac{3}{2}$$

Figure 1.11 Chromatic information and graph entropy.

sumes a fixed (uniform) probability distribution over V, whereas $H(G, \boldsymbol{P})$ allows the probability distribution over V to vary. Figure 1.11 illustrates the relationship between $I_c(G)$ and $H(G, \boldsymbol{P})$.

The two entropy-based measures of graph complexity, $I_a(G)$ and $I_c(G)$, discussed in this chapter capture different aspects of graph structure. Colorings and symmetries of a graph do not necessarily say much about each other. The difference can be seen from examples such as the cycle C_n on n vertices. This graph has a transitive automorphism group so $I_a(C_n) = 0$ for all n, whereas the cycle has chromatic number two or three, depending on whether n is even or odd, and $I_c(C_n) \approx \log 2$. The divergence between the two measures is unbounded in the case of trees that have chromatic number two but (for $n \geq 7$) can have a trivial automorphism group. In these cases, $I_c(C_n) \leq 1$ but $I_a(C_n) = \log n$.

The foregoing observations support the view that structural complexity is in the eye of the beholder. No single measure can capture all aspects of a graph.

Acknowledgment

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement No. W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either express or implied, of the U.S. Army Research Laboratory, the U.S. government, the U.K. Ministry of Defence or the U.K. government. The U.S. and U.K. governments are authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation hereon.

References

1 Arvind, V. Algebra and Computation. Lecture notes (transcribed by Ramprasad Saptharishi), 2007. http://www.cmi.ac.in/~ramprasad/lecturenotes/algcomp/tillnow.pdf, last viewed 5-30-2008.

2 Bent, G., Dantressangle, P., Vyvyan, D., Mowshowitz, A., and Mitsou, V. A dynamic distributed federated database. Proceedings of the Second Annual Conference of the International Technology Alliance, Imperial College, London, September 2008.

3 Biggs, N.L. *Algebraic Graph Theory*. Cambridge University Press, Cambridge, 1993.

4 Brandes, U. A faster algorithm for betweenness centrality. *J. Math. Sociol.* 25 (**2001**), pp. 163–177.

5 Chao, C.C. A note on the eigenvalues of a graph. *J. Combinator. Theory* (Series B) 10 (**1971**), pp. 301–302.

6 Collatz, L., and Sinogowitz, U. Spektrum endlicher Graphen. *Abh. Math. Sem. Univ. Hamburg* 21 (**1957**), pp. 63–77.

7 Everett, M. G., and Borgatti, S. Calculating Role Similarities: An Algorithm that helps

determine the Orbits of Graph. *Social Networks* 10 (**1988**), pp. 77–91.

8 Freeman, L. C. A set of measures of centrality based on betweenness. *Sociometry* 40 (**1977**), pp. 35–41.

9 Hage, P. and Harary, F. Eccentricity and centrality in networks. *Social Networks* 17 (**1995**), pp. 57–63.

10 Khinchin, A.I. *Mathematical Foundations of Information Theory*. Dover Publications, New York, 1957.

11 Korner, J. Coding of an information source having ambiguous alphabet and the entropy of graphs. *Transactions of Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*. (**1971**), pp. 411–425.

12 Lauri, J. and Scapellato, R. *Topics in Graph Automorphisms and Reconstruction*. Cambridge University Press, Cambridge, 2003.

13 Mowshowitz, A. Entropy and the complexity of graphs: I. An index of the relative complexity of a graph. *Bull. Math. Biophys.* 30 (**1968**), pp. 175–204.

14 Mowshowitz, A. Entropy and the complexity of graphs: II. The information content of digraphs and infinite graphs. *Bulletin of Mathematical Biophysics* 30 (**1968**), pp. 225–240.

15 Mowshowitz, A. Entropy and the complexity of graphs: III. Graphs with prescribed information content. *Bull. Math. Biophys.* 30 (**1968**), pp. 387–414.

16 Mowshowitz, A. Entropy and the complexity of graphs: IV. Entropy measures and graphical structure. *Bull. Math. Biophys.* 30 (**1968**), pp. 533–546.

17 Mowshowitz, A. The group of a graph whose adjacency matrix has all distinct eigenvalues. In F. Harary, editor, *Proof Techniques in Graph Theory*, pp. 109–110. Academic Press, New York, 1969.

18 Mowshowitz, A. Graphs, groups and matrices. In *Proceedings of the Canadian Mathematical Congress* (**1971**), pp. 509–522.

19 Mowshowitz, A., Mitsou, V., and Bent, G. Models of network growth by combination. Proceedings of the Second Annual Conference of the International Technology Alliance, Imperial College, London, September 2008.

20 Papireddy, Y.S. A cost efficient approach for finding the orbits to calculate the Entropy of a graph. Project Presentation, Department of Computer Science, The City College of New York, Spring 2007.

21 Rashevsky, N. Life, information theory, and topology. *Bull. Math. Biophys.* 17 (**1955**), pp. 229–235.

22 Sabidussi, G. The centrality index of a graph. *Psychometrika* 31 (**1966**), pp. 581–603.

23 Shimbel, A. Structural parameters of communication networks. *Bull. Math. Biophys.* 15 (**1953**), pp. 501–507.

24 Simonyi, G. Graph entropy: a survey. In W. Cook and L. Lovasz, editors, *Combinatorial Optimization*. DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, 1995.

25 Thompson, D.W. *On Growth and Form* (abridged version edited by J.T. Bonner). Cambridge University Press, Cambridge, 1961.

26 Trucco, E. A note on the information content of graphs. *Bull. Math. Biophys.* 18 (**1956**), pp. 129–135.

27 Wikepedia. Kolmogorov complexity. http://en.wikipedia.org/wiki/Kolmogorov_complexity, last viewed 5-28-08.