Canonical Labeling to Facilitate Graph Comparison78

Carter Butts

Social and Decision Sciences and Center for the Computational Analysis of Social and Organizational

Systems

Carnegie Mellon University

Kathleen M. Carley

Social and Decision Sciences, Center for the Computational Analysis of Social and Organizational
Systems, and H.J.Heinz III School of Policy and Management
Carnegie Mellon University

Abstract

A wide range of social structures can be represented as directed graphs; in order for these graphs to be compared, however, they must often be labeled in a theoretically meaningful fashion. This requirement is rarely met in practice, thus there is a need for an algorithmic approach for labeling and aligning graphs. In particular, a canonical labeling algorithm is sought which allows for the statistical treatment of underlying structural properties. Here, we examine three approaches to this problem: a recursive color-splitting algorithm, a nodal degree ordering algorithm, and an algorithm which orders vertices based on a weighted degree function. These methods are compared against both a random baseline and an optimal criterion under monte carlo simulations. It is demonstrated that labelings under all three methods outperform random labelings under Hamming distance measures, and particular advantages of the recursive color splitting algorithm for the identification of isomorphic graphs are discussed.

Keywords: canonical labeling algorithm, graph comparison, Hamming distance, isomorphism, graph coloring

[†] This work was supported in part by the Center for the Computational Analysis of Social and Organizational Systems and the Institute for Complex Engineered Systems at Carnegie Mellon University.

The authors would like to thank David Krackhardt, Pat Doreian, John Skvoretz, Isa Verdinelli, and Jiayin Xiang for their input and encouragement.

^{*} This material is based upon work supported under a National Science Foundation Graduate Fellowship.

This work was supported by Grant No. N00014-97-1-0037 from the Office of Naval Research (ONR), United States Navy.

Introduction and Motivation

Even a brief perusal of the empirical literature on social networks suggests a great deal of diversity in human social structures; this diversity is present not only in informal networks (e.g., friendship relations, acquaintance chains) or but also in more constrained structures (e.g., familial ties, authority or reporting relations). Particularly when considered as sociograms or raw sociomatrices, many networks¹ appear to be quite distinct. This raises some key questions: in particular, we require a formal means of establishing whether or not two or more given structures are truly different, and of quantifying this difference when present. Similarly, we may wish to determine a formal standard for the identification of changes in social structure - a problem which ultimately amounts to the same comparison considered above.

One means of answering these sorts of questions is lies in a descriptive index approach. As noted, many aspects of social structure can be represented as sets of networks or graphs with relationships among these graphs. Among such networks might be those representing communication or authority relations, resource access relations, mutual affinity relations, kinship ties, and so forth. Once a particular structure of interest has been identified, then each of these sub-structures can be measured and represented as a network or graph. Using a variety of descriptive indices (e.g., density, span of control, centralization, reciprocity), differences or similarities between these graphs can be measured. By systematically comparing these indices across structures, it should be possible to classify and/or to discriminate between them.

Here we use the terms network and graph interchangeably. However, in some literatures the term "graph" is restricted to binary (and symmetric) matrices and networks to weighted matrices; in this usage, the majority of the "networks" we consider would be referred to as directed graphs.

Unfortunately, there does not currently exist a commonly accepted taxonomy for classifying social structures. Even in the study of organizations - where one might expect natural constraints to facilitate such an endeavor - debate rages over whether or not such a taxonomy is possible, let alone useful. McKelvey (1982) sees a need for such a taxonomy. Hannan and Freeman (1989), by contrast, argue that categories of organizational designs should be specified according to the interests of the researcher. Some schemes for classifying organizations have been based on strategy (Romanelli, 1989b) or product service (Fligstein, 1985). Other researchers have classified organizations using multiple dimensions. For example, Aldrich and Mueller (1982) categorize organizations using the dimensions of technology, coordination, and control. In contrast with these previous efforts, what we wish to suggest is a graph theoretic approach to this problem. Specifically, we here conceptualize social structures as sets of associated graphs. We then attempt to develop a method of distinguishing alternative, and possibly new, configurations by locating those structures which differ (statistically) significantly from others on a metric distance measure, under a simple null hypothesis. Particular attention is paid to developing a graph theoretical procedure for locating common networks and sub-networks. This is critical, as the ability to locate a common network is necessary for developing a mathematical criterion for determining whether the measures of structure for two different graphs are meaningfully different. In order to determine whether the difference between structures is significant we will need to be able to define the distributions of sets of graphs under a null hypothesis and measure the distance between graphs. Thus, by being able to define distributions of sets of graphs in terms of such concepts as their central tendency, we hope to answer fundamental methodological and theoretical questions regarding social architectures.

Problem Background

Network measures can be used to characterize graphs; similarly, statistically significant differences in these indices can be used to indicate differences in structure. At the network level, measures such as density, hierarchy, and graph connectivity are available for characterizing graphs (Krackhardt, 1994), (Wasserman and Faust, 1994). While most of these indices can be applied to any data which can be represented as graphs, whether or not they are meaningful depends on the data's substantive associations. For example, while span of control makes sense for directed graphs representing command structures, it makes less sense for graphs representing friendship ties. Each aspect of a given social structure which can be represented as a graph will then have its own set of descriptive indices; clearly, we can attempt to contrast graphs on the basis of differences in these measures. We illustrate this approach in Figure 1. In Figure 1, eleven hypothetical social structures are shown, and their ratings on a number of dimensions are indicated².

<Insert: Figure 1. Illustrative Structures and Measures>

It is not our intent in Figure 1 to exhaustively list all descriptive indices that are possible for these graphs. Rather, we have merely illustrated a few types of measures possible for only one aspect of the reporting structure. The first point we wish to make is that this descriptive index based method is a possible approach for characterizing differences between graphs. Exactly what the appropriate indices are for any given each aspect of a social structure and whether or not any of these indices are predictive of any behaviors of interest is a separate issue. Indeed, we expect that many social networks with very different outcome characteristics will be identical on some

dimensions and different on others -- hence, the predictive power of the descriptive index approach will vary depending on which indices are examined and upon which of the substructures the indices are based. Consider, for example, two organizational forms that have identical command structures but different task precedence orderings; though they will almost certainly exhibit different performance characteristics, a cursory analysis focusing only on authority relations might predict the opposite. Of course, a comparison of social structures focusing on the correct set of indices and of substructures may well be highly predictive; this does not, however, detract from our second point that a general and reliable comparison of social structure requires a characterization of the underlying distribution of these graphs.

As we have indicated, graphs with widely disparate configurations can look very similar given a set of network measures. This makes statistical comparison difficult. For example, all of the graphs in Table 1 are distinct structures; yet, each pair appears identical on a number of dimensions. On the one hand, this problem can be (in theory) resolved by using a suite of measures for each structure that cover the range of implicit dimensions. Assuming that all relevant dimensions have been characterized, structures that are meaningfully different should show up as different on one or more of these measures. The difficulty here, however, lies in defining a set of indices that exhaust all possibilities and so ensure coverage of all of the ways in which the graphs could differ. Further, this approach is potentially problematic in that it requires not only knowledge of a wide range of network indices, but also of their distributions and possible associations (e.g., only certain levels of reciprocity are available at a given level of network density). At present, such a library of measures and distributions is incomplete at best, casting doubt on the efficacy of this method.

² All ties for the illustrative structures are considered to be symmetric.

An alternative, structural, approach is to map two graphs onto each other and then to look for discrepancies in their overall structures. For the special case in which the two graphs to be compared are uniquely labeled, such a difference between them can be readily captured by the Hamming distance (Hamming, 1950). Note that any dichotomous digraph can be equivalently represented as a binary matrix M with the number of rows/columns equal to the number of nodes³ and with cell values of $M_{i,j}$ =1 representing the presence of the link from node i to node j and $M_{i,j}$ =0 indicating the absence of such a link. The Hamming distance is simply the minimum number of cells whose value needs to be flipped so that the two matrices come to be identical; i.e., the minimum number of links that need to be added/dropped to make the two graphs identical. In Figure 2, the matrices corresponding to several of the networks in Figure 1, and their Hamming distances from each other are shown. Each of these matrices is an adjacency matrix indicating those edges or links which are present between nodes (all edges are assumed to be directed from the lower level to the upper level). Note that, in order to create Figure 2, it was necessary first to order (or label) the nodes of each digraph; here this labeling was accomplished using the proposed color-splitting algorithm described below.

<Insert: Figure 2. Hamming Metric and Four Sample Structures>

Previous work has demonstrated that, for sets of graphs in which all nodes are labeled, it is possible to derive a structural distribution and to identify its central graph. Banks & Carley (1994) developed a non-parametric network based statistical technique for identifying the central graph⁴, the standard deviation of distances to that graph, and the confidence intervals about it.

³ In the N-modal case, each dimension represents a given set of actors. Generalization of the Hamming distance to this case is unproblematic.

⁴ We use here the term central graph as we want to emphasize the relationship between this graph and the mean that one gets for variable level data. In other contexts, the terms consensus structure (Krackhardt, 1987), cultural

The central graph, which we shall consider at greater length presently, is that network containing the union of the node sets of the graphs from which it is constructed and in which two nodes are adjacent if and only if they have been adjacent in 50% or more of the graphs in the set5. Whether or not a graph is significantly different than the central graph is assessed by comparing its Hamming distance from the central graph with that which would be expected under the null hypothesis. Currently, it is possible to determine whether or not two graphs are significantly different only for the special case in which all nodes are labeled (each node has a unique identifier) and in which both graphs share the same label set. This technique further assumes a null hypothesis in which all links between nodes are independent and identically distributed (Banks and Carley, 1994). Given these assumptions, it is possible to generate a distribution of networks from the sample population using non-parametric bootstrapping techniques, determine the first moment of this distribution by identifying the central graph, and then calculate the distribution of distances from this central graph using the Hamming metric. A non-parametric test of means can then be used to determine whether the distances of the networks in question from the central graph are sufficiently large to reject the hypotheses that they are the same as the central graph. The central graph, hence, is the network equivalent of a mean for a non-network variable.

In many situations, however, one is forced to work with graphs in which nodes are not labeled, or for which the labels are not relevant. For example, in one work team two roles might be labeled Systems Analyst and Team Coordinator; whereas in another these might be labeled Systems Engineer and Group Leader. These differences in labels may be due simply to disparate

consensus (Romney, Weller and Batchelder's, 1986), and majority intersection structure (Carley, 1984, 1986) have been used to denote the same basic idea.

In other words, it is the mean graph, dichotomized at the 50% level.

documentation standards and may not reflect real underlying structural differences. Given a set of unlabeled graphs it might still be possible to produce a partial labeling of nodes by "coloring" them: one might say that those nodes possessing some given characteristic are colored yellow, while those nodes with another characteristic are red, and so on. For example, in a world system context one might define all nodes representing core states as yellow and all nodes representing members of the periphery as blue. Regardless, given a set of unlabeled graphs, or at best colored graphs, we might still want to identify any central tendencies present among them.

For colored networks, however, there are many ways in which two networks can overlay, thus complicating the process of locating the central graph. For example, in Figure 3, a1 and a2 are the same color and therefore interchangeable. Two different matches may be found (simply in terms of node a) by either lining up the a1's and a2's or by lining up the left sides and the right sides. Still other matches are possible when the nodes of other colors are considered; thus, when unique labelings are not available, there are multiple ways in which the central graph can be calculated.

<Insert: Figure 3. Two Colored Networks>

One approach to the comparison of unlabeled (or partially labeled) graphs is to re-label the nodes based on their network properties. Once a unique and complete labeling has been established, it is possible to use traditional methods for the assessment of structural distance. The choice of algorithm for labeling the nodes is critical, however: the reason for this is that the Hamming metric is sensitive to minor permutations of nodal labels. An example of this phenomenon may be seen in Figure 4. For these graphs the three comparisons differ from the minimum distance of 0 to the maximum distance of 10 - despite the fact that all six graphs are

perfectly isomorphic! This phenomenon suggests that using arbitrary label choices may lead to poor inferences regarding structural distance: a critical problem for cases in which not all nodes (e.g., people, resources or tasks) are interestingly unique (that is, non-interchangeable in terms of the theory of interest). Imagine, for instance, two workgroups, each with a group leader, L, two analysts (A & B), and two tasks (t1 & t2) such that in the first workgroup A works on t1 and B works on t2 whereas, in the second workgroup, A works on t2 and B works on t1. Though these workgroups are functionally identical, a straightforward application of the Hamming metric would indicate a structural difference between the two. In order to correctly assess the difference between the two graphs, then, nodes of the same "color" (such as, in this case, analysts working with the same group leader) must be treated as interchangeable; treating the structure as a conventional, pre-labeled graph is not a viable option.

<Insert: Figure 4. Hamming Distances for Permuted Matrices>

If arbitrary labels yield arbitrary distances, what sort of labelings might prove more useful?

In general, it would seem reasonable to seek a method of labeling nodes such that A) the

Hamming distance between labelings of any two graphs will remain constant across (pre-labeled)

permutations of those graphs, and B) the Hamming distance between any two labeled graphs will

be minimized⁶. Finding a general method of achieving this goal, however, poses several

problems. The first is simply one of combinatorics: for an unlabeled graph with N nodes, the

number of possible labelings is equal to the number of node permutations⁷, or N!. While, in

theory, one could exhaustively search the space of labelings for the one which minimizes the

⁶ This follows from the fact that A) we would prefer for isomorphic graphs to have a Hamming distance of 0, and B) the Hamming distance can never fall below the minimum number of tie additions/subtractions needed to make the two graphs under comparison isomorphic. By minimizing the Hamming distance, we ensure that our comparison is as close as possible to the smallest number of changes required to convert one graph into another.

More generally, permutations, complete node orderings, complete degenerate colorings, and complete labelings of the node set are all equivalent in this context.

Table 6: Observed Hamming Distance.	for Isomorphism Experiment
-------------------------------------	----------------------------

		3 - control for assured present 2steperiment					
Method	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	
Rand	0	22	70	107.5	162	496	
WDeg	0	16	64	101.7	158	478	
Degree	0	4	28	53.97	72	437	
RCS 1	0	0	0	8.183	4	204	
RCS 2	0	0	0	3.078	0	212	
RCS 3	0	0	0	2.787	0	242	
RCS 4	0	0	0	2.694	0	196	
RCS 5	0	0	0	2.841	0	158	

Given two such labeled digraphs, we may define a metric distance between them as per Hamming (1950). Let $\mathbf{H}_1 = L_1(\mathbf{G}_1)$ and $\mathbf{H}_2 = L_2(\mathbf{G}_2)$ with vertex sets such that $V_j = V_2 = V_U$. (This does not imply loss of generality, since isolates can be added to vertex sets as necessary.) Now, define an indicator function $\delta_h(e(v_i, v_j))$ such that

$$[1] \quad \delta_h(e(v_i, v_j)) = \begin{cases} 1 & if \ \exists \ e(v_i, v_j) \in \mathbf{H_h} \\ 0 & otherwise \end{cases}$$

The function δ permits us to count directed edges within a given labeled digraph. To derive the Hamming distance between two labeled graphs, then, we simply count the number of directed edges which exist in one graph and not the other. This gives us the following expression for the Hamming distance:

[2]
$$D(\mathbf{H}_1, \mathbf{H}_2) = \sum_{i=1}^{|V_i|} \sum_{i=1}^{|V_j|} |\delta_1(e(v_i, v_j)) - \delta_2(e(v_i, v_j))|$$

The metric distance given by [2] permits us to characterize the difference between two given labeled graphs. This distance cannot, however, be directly applied to unlabeled graphs due to the fact that unique identification of vertices is necessary to count the number of disperate edges. How, then, can we proceed to characterize the distance between two unlabeled graphs? As a starting point, we observe that

[3]
$$D_o(\mathbf{H}_1, \mathbf{H}_2) = D_o(L_1(\mathbf{G}_1), L_2(\mathbf{G}_2))$$

where we use D_0 to refer to the "observed" Hamming distance between G_1 and G_2 (the reason for this change in nomenclature will become obvious presently). Note that, as [3] makes particularly clear, the observed Hamming distance between two digraphs depends both on their underlying unlabeled graphs and on the labelings applied to those digraphs. To remove this

comparisons per second would take over 77,000 years; this is probably a bit long to wait for a single data point.

dependency, it is worthwhile to consider some function which does not involve both components; thus, we define the following notion of structural distance:

[4]
$$D_s(\mathbf{G}_1, \mathbf{G}_2) = \min_{\forall t_1, t_2} (D_o(L_1(\mathbf{G}_1), L_2(\mathbf{G}_2)))$$

Because D_S reflects the minimum distance between any labeled digraphs from the set of all labelings on G_1 and G_2 , it depends only on our choice of unlabeled graphs – in that sense, then, it can be said to characterize an entirely structural distance between graphs. Further, because D_S is defined as the D_O produced by a particular choice of labelings, D_S is also a direct generalization of the Hamming distance, and is itself a metric⁹. The definitions from [3] and [4] permit us to define yet another quantity, the *labeling distance*:

[5]
$$D_L(L_1(\mathbf{G}_1), L_2(\mathbf{G}_2)) = D_O(L_1(\mathbf{G}_1), L_2(\mathbf{G}_2)) - D_S(\mathbf{G}_1, \mathbf{G}_2)$$

which is clearly nothing more than the "extra" observed Hamming distance between two digraphs due to the choice of labeling. The form of [5], then suggests the following decomposition:

[6]
$$D_o(L_1(\mathbf{G}_1), L_2(\mathbf{G}_2)) = D_s(\mathbf{G}_1, \mathbf{G}_2) + D_L(L_1(\mathbf{G}_1), L_2(\mathbf{G}_2))$$

As [6] shows, we may express the observed Hamming distance between two labeled digraphs as the sum of the structural distance between those graphs' underlying labeled graphs and the labeling distance induced by our particular selection of labels for the vertices of the graphs in question. Furthermore, [6] allows us to make the following observation, which will prove to be of great importance: if two labeled graphs may be relabeled in such a way that the labeling distance between them is reduced to zero, then the observed Hamming distance between the

Clearly, $D_S(G_i,G_j)=0$, $D_S(G_i,G_j)=D_S(G_j,G_i)$, and $D_S(G_i,G_j)>0$ \forall i.j. The triangle inequality $D_S(G_i,G_j)+D_S(G_j,G_k)\geq D_S(G_i,G_k)$ \forall i.j.k follows from the fact that edge additions/deletions may be performed in any order, and thus one can always map G_i into G_k by way of G_j if that results in a lower Hamming Distance.

relabeled graphs will be equal to the structural distance between their respective unlabeled graphs. While we cannot directly observe the structural distance between two unlabeled digraphs, then, we may nevertheless infer it from the observed Hamming distance between digraphs that have been labeled in the appropriate fashion.

While the decomposition of [6] is an important tool, it does beg a difficult question: how, exactly, should graphs be labeled in order to ensure that the labeling distance between the graphs will be equal to zero? This is not a trivial question; the number of unique labelings of any given graph is equal to IV!, and it is a priori nonobvious how to identify those labelings which will minimize the Hamming distance between two graphs. Indeed, the problem is somewhat more difficult even than the above implies: in order for the decomposition of [6] to permit comparisons among more than two graphs at any given time, we require a general mechanism of labeling graphs so as to cause the labeling distance to be zero for all comparisons 10. Such a mechanism constitutes a canonical labeling algorithm, and may be more formally specified as follows 11:

Definition (Canonical Labeling Criterion): Let a labeling algorithm L_C be said to satisfy the canonical labeling criterion * if and only if $D_O(L_C(G_i),L_C(G_j))=D_S(G_i,G_j)$ \forall Gi, Gi.

Definition (Canonical Labeling Algorithm): Let a labeling algorithm L_C be said to be a canonical labeling algorithm if and only if it satisfies *.

As the above definitions make clear, we are seeking a labeling algorithm which causes observed Hamming distances to be equal to structural distances for all graphs so labeled (an equivalent

¹⁰ Thus, an adaptive search to find the minimal Hamming distance between two digraphs would not qualify, although

such a method is useful for a number of related problems.

Throughout this paper, all references to canonical labeling algorithms concern those meeting the criteria specified below. However, the term may be more generally used to refer to any algorithm which expresses unlabeled graphs in a canonical labeled form, whether or not Hamming distances between labeled graphs are minimized.

condition to the minimization of D_L). Does such an algorithm exist? In order to verify this proposition, we must first prove some preliminary results:

Lemma 1 (Choice of Labeling): Given some fixed labeling $L_1(G_1)=H$, \exists an $L_2(G_2)$: $D_0(H,L_2(G_2))=D_S(G_1,G_2)$,

 $\label{eq:proof: Let l_1 and l_2 be labelings of G_1 and G_2: $D_O(l_1(G_1),l_2(G_2)) = D_S(G_1,G_2)$. Let R be a relabeling such that $R(l_1(G_1)) = L_1(G_1) = H$. Choose L_2: $L_2(l_2(G_2)) = R(l_2(G_2))$. Then$

$$D_{O}(H,L_{2}(G_{2})) = D_{O}(R(l_{1}(G_{1})),R(l_{2}(G_{2}))) = \sum_{i=1}^{|V_{i}|} \sum_{j=1}^{|V_{i}|} \left| \delta_{1}\left(e\left(v_{R(i)},v_{R(j)}\right)\right) - \delta_{2}\left(e\left(v_{R(i)},v_{R(j)}\right)\right) \right|$$

$$=D_O(I_1(G_1),I_2(G_2))=D_S(G_1,G_2)$$

Lemma 1 simply demonstrates that it is not necessary to control the labelings of both graphs simultaneously to find the minimal Hamming distance between two digraphs. This result is an important one for our subsequent argumentation; also important is Lemma 2:

Lemma 2 (Failure Condition): If and only if the labeling algorithm L_C fails to satisfy *, then \exists G_1 , G_2 , G_3 , G_4 : $D_O(L_C(G_1),L_C(G_2))=D_S(G_1,G_2) \Rightarrow D_O(L_C(G_3),L_C(G_4))>D_S(G_3,G_4)$.

$$\begin{split} \textbf{Proof:} & \text{ (Necessity) Assume that } \not\exists \ G_1, \ G_2, \ G_3, \ G_4 \text{: } D_O(L_C(G_1), L_C(G_2)) = D_S(G_1, G_2) \Rightarrow \\ & D_O(L_C(G_3), L_C(G_4)) > D_S(G_3, G_4). \quad \text{Then } D_O(L_C(G_3), L_C(G_4)) = D_S(G_3, G_4) \ \forall \ G_3, G_4, \ \text{which} \\ & \text{ (by definition) implies that } L_C \text{ satisfies *}. \end{split}$$

(Sufficiency) If \exists G_1 , G_2 , G_3 , G_4 : $D_O(L_C(G_1),L_C(G_2))=D_S(G_1,G_2) \Rightarrow$ $D_O(L_C(G_3),L_C(G_4))>D_S(G_3,G_4)$, then it is not true that $D_O(L_C(G_3),L_C(G_4))=D_S(G_3,G_4) \forall$ G_3 , G_4 , and thus L_C cannot satisfy *. \blacksquare

Lemma 2 shows that the canonical labeling criterion can be fulfilled if and only if there is no case for which there are mutually exclusive optimal labelings. This, combined with Lemma 1, gives us the basis for our proof of the existence of a canonical labeling algorithm as we have defined it.

Theorem 1 (Existence of a Canonical Labeling Algorithm): \exists a labeling algorithm $L_C: L_C$ satisfies *.

Proof: Assume that \exists a labeling algorithm L: L does not satisfy *. By Lemma 2, \exists G₁, G₂, G₃, G₄: $D_O(L(G_1),L(G_2))=D_S(G_1,G_2) \Rightarrow D_O(L(G_3),L(G_4))>D_S(G_3,G_4)$. Now, let $L(G_1)=H_1$ and $L(G_2)=H_2$ and modify L as follows:

- i) If $G_3=G_4$, fix $L(G_3)=H_3$. By substitution, $L(G_4)=L(G_3)=H_3$. From the metric property of the Hamming distance, $D_0(H_3,H_3)=0=D_0(L(G_3),L(G_4))$, and $D_5(G_3,G_4)=0$. \therefore L fails the criterion of Lemma 2, and thus satisfies *.
- ii)If $G_3 \neq G_4$, $G_1 = G_3$, and $G_2 = G_4$ fix $L(G_3) = H_3$. By our earlier assumption, $D_O(L(G_1),L(G_2)) = D_O(H_1,\ H_2) = D_S(G_1,G_2).$ Substituting for G_3 and G_4 , we can see that $D_O(L(G_3),L(G_4)) = D_O(L(G_1),L(G_2)) = D_O(H_1,\ H_2) = D_S(G_1,G_2) = D_S(G_3,G_4)$. Thus, L fails the Lemma 2 criterion and must satisfy *.
- iii) If $G_3 \neq G_4$, $G_1 = G_4$, and $G_2 = G_3$, we substitute as above. Thus, we can see that $D_O(L(G_3),L(G_4)) = D_O(L(G_2),L(G_1)) = D_O(H_2, \quad H_1) = D_S(G_2,G_1) = \quad D_S(G_3,G_4). \quad L$ therefore fails the Lemma 2 criterion and satisfies *.
- iv) If $G_3 \neq G_4$, $G_1 = G_4$, and $G_2 \neq G_3$, substitute to obtain $L(G_4) = H_1$. By Lemma 1, we can choose H_4 : $D_0(H_1, L(G_4)) = D_5(G_3, G_4)$. By the Lemma 2 criterion, then, L satisfies *.

v)If $G_3 \neq G_4$, $G_1 = G_3$, and $G_2 \neq G_4$; if $G_3 \neq G_4$, $G_2 = G_3$, and $G_1 \neq G_4$; or if $G_3 \neq G_4$, $G_2 = G_4$, and $G_1 \neq G_3$, we substitute and employ Lemma 1 as per (iv) above.

vi) If $G_3 \neq G_4$, $G_1 \neq G_3$, $G_1 \neq G_4$, $G_2 \neq G_3$, and $G_2 \neq G_3$, fix $L(G_3) = H_3$. By Lemma 1, we can choose H_4 : $D_O(H_1, L(G_4)) = D_S(G_3, G_4)$. This causes L to fail the Lemma 2 criterion, and thus to satisfy *.

As the above cases cover all choices of G_1 , G_2 , G_3 , G_4 , we conclude that the condition of Lemma 2 is not satisfied \forall L; \therefore \exists some labeling algorithm L_C : L_C satisfies *.

Theorem 1 above demonstrates that there exists some canonical labeling algorithm; it still leaves many questions unanswered, however. Given that some L_C exists, what can we say about it? While Theorem 1 could be used to construct a functional algorithm, such an algorithm would be computationally infeasible. This follows from the fact that the substitutions relied upon in the above proofs require the identification of labeled graphs with their isomorphism classes: while this is mathematically reasonable, it imposes the computational requirement of searching through the space of graph labelings. As noted, the cardinality of such a space is equal to IVI! for any given graph, making a standard search prohibitive. Adaptive techniques might be applied to obtain a heuristic solution, but the difficulty of the search task suggests that this may be computationally difficult as well. In this paper, then, we will not pursue this particular avenue in attempting to identify a canonical labeling algorithm.

Another set of important issues concerns the effects of canonical labeling on other tools for network comparison, such as the central graph. As has been noted, the central graph is derived from Hamming distance measurements, and as such can only be calculated from labeled graphs. While this poses no difficulties for analyses in which graphs have been prelabeled in a

theoretically meaningful fashion, this is not the case (as we have seen) for arbitrarily labeled structures. Will our canonical labeling algorithm permit us to find a central structure which is meaningful for unlabeled graphs? To see this, we first must define the central graph in terms of the observed Hamming distance. Following Banks and Carley (1994):

Definition (Central Graph): Given a set of labeled digraphs \mathbf{H}_1 , \mathbf{H}_2 , ... \mathbf{H}_n defined on the vertex set V_U , let the central graph be that labeled digraph $\mathbf{H}_{\mathbf{C}}$ which minimizes $\sum_{i=1}^n \left(D_o(\mathbf{H}_{\mathbf{C}},\mathbf{H}_i)\right)^2.$

From this definition, then, we may substitute to find the expression for the sum minimized by the central graph in terms of labelings of unlabeled graphs.

[7]
$$\sum_{i=1}^{n} \left(D_o(\mathbf{H}_{\mathbf{C}}, \mathbf{H}_{\mathbf{i}}) \right)^2 = \sum_{i=1}^{n} \left(D_o(L_{cg}(\mathbf{G}_{\mathbf{C}}), L_i(\mathbf{G}_{\mathbf{i}})) \right)^2$$
$$= \sum_{i=1}^{n} \left(D_s(\mathbf{G}_{\mathbf{C}}, \mathbf{G}_{\mathbf{i}}) + D_L(L_{cg}(\mathbf{G}_{\mathbf{C}}), L_i(\mathbf{G}_{\mathbf{i}})) \right)^2$$

Recall, however, that by definition a canonical labeling algorithm requires that D_L =0 for all graph comparisons. Therefore, for the case in which the set of labeled digraphs H_1 , H_2 , ... H_n are relabeled using a canonical labeling algorithm L_C , [7] reduces to

[8]
$$\sum_{i=1}^{n} \left(D_{S}(\mathbf{G}_{C}, \mathbf{G}_{i}) + D_{L}(L_{C}(\mathbf{G}_{C}), L_{C}(\mathbf{G}_{i})) \right)^{2} = \sum_{i=1}^{n} \left(D_{S}(\mathbf{G}_{C}, \mathbf{G}_{i}) \right)^{2}$$

which is precisely the quantity which would be minimized by a central graph defined only on the underlying, unlabeled graphs. Clearly, then, one useful side effect of the canonical labeling algorithm is the fact that it permits the generalization of the central graph to cases in which the digraphs to be examined do not possess theoretically meaningful labels.

The above discussions have centered on two extreme possibilities: either a digraph is conceived of as being uniquely labeled, or it is considered to have no labels at all. As has been indicated, however, it is often the case in network analysis that certain groups of vertices may be considered to be interchangeable without permitting interchange among all vertices. Such graphs may be said to be *partially prelabeled*, or *nondegenerately colored*, and it is reasonable to question whether the above results can be applied to such graphs. As it happens, the definitions of structural distance, labeling distance, and central graph can be unproblematically extended to cover nondegenerately colored digraphs¹², and the results based on these definitions (including the theorems given above) should hold for this larger class of graphs. Formal extension of these results will not be given here, although an outline of the process is suggested in the above note.

To summarize the results of this section, we may express the observed Hamming distance between two labeled digraphs as the sum of the structural distance between the digraphs' respective unlabeled counterparts and the surplus distance induced by the choice of labeling. A canonical labeling algorithm of the type we consider here reduces this labeling distance to zero for all pairs of graphs, thus permitting us to infer the structural distance between unlabeled graphs from the observed Hamming distance between unlabeled graphs. A secondary implication of this fact is that central graphs defined on canonically labeled sets of graphs correspond to the central graphs of their respective unlabeled graphs. Though we have proven the existence of a canonical labeling algorithm with the above properties, we have not yet identified one that is computationally feasible (nor have we shown that such exists). In the sections which follow, then, we shall examine several heuristic alternatives which fulfill the canonical labeling criterion to varying degrees. Our goal shall be to identify, if possible, one or

¹² This process starts by defining limitations on the set of possible labelings, and redefining the terms of the

more algorithms which serve reasonably effectively as canonical labeling algorithms, and which are sufficiently computationally facile to be serviceable in a typical research context. We shall also consider, by the same token, the estimated error produced by not employing a canonical labeling algorithm, and shall examine the effects of our proposed alternatives on the distribution of distances about the central graph.

Three Approaches to the Labeling Problem

There is no known technique for locating the optimal 13 match between two colored networks, let alone locating the central structure based on the optimal match on a set of structures, which does not involve exhaustive search procedures. Heuristic approaches to this general problem canonical labeling algorithms - have been developed by researchers working in a number of disciplines, including computer science, electrical engineering, chemistry, mathematics, and physics. This previous work has focused on the discovery of canonical labeling algorithms for isomorphism testing ((Luks, 1982), (Babai & Luks, 1983)), graph identification ((Cai et al, 1992)), characterization of regular graphs (Kučera, 1987), constructive enumeration of structures ((Pospichal & Kvasnička, 1996), (Read, 1978), (Skvoretz, 1996)), and representation of canonical transformations (Lim, 1989). Unfortunately for our purposes, these developments have tended to focus on true graphs (as opposed to directed or multigraphs), regular graphs, and graphs of restricted valence. Social networks do not generally satisfy these strict conditions in theory or in practice, and hence there is some difficulty in applying some of the previous results

decomposition in [6] using these new definitions. The other results follow.

That is, one which satisfies the minimal Hamming distance criteria; henceforth, these comprises our definition of "optimal".

to the present problem¹⁴. Furthermore, our task is a bit more complicated than simply mapping graphs to their isomorphism classes: we must minimize the Hamming distance between labeled graphs *in general*, and minimize the mean Hamming distance to the central graph of an arbitrarily chosen graph set¹⁵. To our knowledge, this has not been previously attempted, nor have existing algorithms been shown to satisfy these requirements.

To approach the particular problem of Hamming distance minimization in social networks, we here describe three "matching" or "alignment" heuristics for producing canonical labelings of graphs, along with a random baseline. These labeling algorithms should work in such a way as to, ideally, locate the optimal match between a set of unlabeled or colored networks; in the terms of our earlier decomposition (see equation [6]), these labeling algorithms should minimize the labeling distance D_L in order to permit the observation of the structural distance D_S from the observed Hamming distance D_O . Once optimal labelings for the networks are determined, the previously described techniques for locating the central graph and determining differences among networks can be applied.

The approach and tools for generating these distributions and locating the central graphs should generalize for networks with other constraints and colorings than those we examine: hypergraphs, for instance can be losslessly transformed into bipartite graphs prior to the application of the labeling process. Likewise, many of the methods used here may be applied, with some modification to valued or weighted graphs. Nevertheless, caution should be exercised when attempting to utilize these methods to analyze structures which are non-dichotomous, or whose structural properties otherwise differ greatly from those considered in the experiments

¹⁴ Although one should note that it *is* possible to losslessly convert irregular graphs to regular ones via a simple algorithm, and that bounded valance is not a problem in the event that the labeling to be derived is to be used on a graph set with a constrained number of nodes (e.g., not an open-ended intertemporal data set).

which follow. Although these algorithms may continue to function on such graphs, we cannot be certain how they will behave; such extrapolations require further testing to be considered completely safe.

The Recursive Color-Splitting Algorithm (RCS)

The first approach to the labeling problem which shall be considered here consists of a recursive algorithm which splits sets of identically colored nodes into maximal subsets of recolored nodes such that all nodes with each set are members of the same color class, and such that no color class contains members in more than one subset. Nominally, this process terminates when no subsets (or, by extension, color classes) with more than one member remain, thus producing a complete graph coloring. (As we shall see, the algorithm will be unable to split certain color sets - the impact of this fact for algorithmic performance will be discussed.) The method by which subsets are recolored is based on an ordering principle which sorts nodes based on both local (e.g., degree) and global (e.g., connection to high-degree alters, number of directed walks to other nodes) network features; hence, the algorithm may be applied to individual graphs, and works independently of any within-color preliminary ordering.

This color-splitting algorithm works by "feeling out" a graph's structure; hence, one of the key elements used in this process is the set of paths between nodes. In particular, we shall be interested in the number of directed walks between nodes in the graph given by adjacency matrix \mathbf{A}^{16} which follow specific trajectories. Formally, let $\mathbf{t}_{v,d}$ represent a trajectory vector consisting

 $^{^{15}}$ Isomorphism testing corresponds to the minimization of D_{L} when $D_{S}\!\!=\!\!0,$ in our notation.

in Throughout this discussion, we shall treat directed graphs and their sociomatrices as interchangeable.

of d elements, each of which is a member of the set $\{0,1\}$, which has value v where value is given by

[9]
$$v = \sum_{i=0}^{d} \mathbf{t}_{vd(i)} 2^{i}$$

Each given v and d, then, describes a unique t (which can be thought of as the binary expression of v in d digits). Each vector t, in turn, represents a particular family of walks on the graph given by A. In particular:

[10]
$$\mathbf{W}_{t_{dr}} = \prod_{i=0}^{d} (\mathbf{t}_{dv(i)} \mathbf{A}^{T} + (\mathbf{I} - \mathbf{t}_{dv(i)}) \mathbf{A})$$

gives us the total number of directed walks following the trajectory vector \mathbf{t}_{dv} for all vertices in \mathbf{A} . As [10] implies, the trajectory parameter d refers to the total length of the trajectory, while the trajectory's value (ν) expresses, via a series of binary digits, a set of "instructions" regarding walks on the ingoing and outgoing edges of \mathbf{A} . This last follows from the scalars of \mathbf{A} and \mathbf{A}^{T} contained within the product term of [10]: when $\mathbf{t}_{df(i)}$ is equal to 0, the product element as a whole is equal to \mathbf{A} (thus tracing outgoing edges), and when $\mathbf{t}_{df(i)}$ is equal to 1, the product element is equal to \mathbf{A}^{T} (tracing ingoing edges).

As has been noted, the basic problem of finding a reasonable labeling rests on using structural properties of the network to produce a unique ordering of nodes. One such property is the geometry of walks on the digraph; another which might be considered in this context is the row sum, or outdegree. (In an authority or reporting structure the outdegree of a node would be the number of others to which each node reports.) Because outdegree follows directly from the graph structure, it is invariant under permutation; furthermore, it is simple to calculate and compare. Unfortunately, however, most networks involve numerous nodes with identical

- 1) Let d=0
- 2) Let v=0
- 3) Given a maximal set of identically colored nodes, G, recolor members as follows:
- 4) (For all i and j in G): If $SR_i(d,v) > SR_j(d,v)$ then SK(i) > SK(j); else if $SR_i(d,v) < SR_j(d,v)$ then SK(i) < SK(j)
- 5) (For all i and j in G such that $SR_1(d,v) = SR_1(d,v)$: If $SC_1(d,v) > SC_1(d,v)$ then SK(i) > SK(j); else if $SC_1(d,v) < SC_1(d,v)$ then SK(i) < SK(j)
- 6) If $(v<2^d-1)$ then (for all i and j in \mathbf{G} such that $\mathrm{SR}_1(d,v)=\mathrm{SR}_j(d,v)$ and $\mathrm{SC}_1(d,v)=\mathrm{SC}_j(d,v)$): let v=v+1 and goto (3) for all subsets $\mathbf{G}_1\ldots\mathbf{G}_m$ such that $\mathrm{SK}(k)=\mathrm{SK}(I)$ for all k and I in \mathbf{G}_n and $\mathrm{SK}(k)\neq\mathrm{SK}(I)$ for all k in \mathbf{G}_n and I in I i
- 7) If $(v=2^d-1 \text{ and } d< N-1)$ then (for all i and j in \mathbf{G} such that $\mathrm{SR}_i(d,v)=\mathrm{SR}_j(d,v)$ and $\mathrm{SC}_i(d,v)=\mathrm{SC}_j(d,v)$): let d=d+1 and goto (2) for all sets $\mathbf{G}_1\ldots\mathbf{G}_m$ such that $\mathrm{SK}(k)=\mathrm{SK}(l)$ for all k and l in \mathbf{G}_n and $\mathrm{SK}(k)\neq\mathrm{SK}(l)$ for all k in \mathbf{G}_n and l in \mathbf{G}_{n+k} ($n-1\leq n\leq m-n$), for all n in l...m
- 8) If $(v=2^d-1 \text{ and } d=N-1)$ then (for all i and j in G such that $SR_i(d,v)=SR_j(d,v)$ and $SC_i(d,v)=SC_j(d,v)$): SK(i)=SK(j)

9) End

(Note that all explicitly set orderings are permanent; i.e., if SK(i)>SK(j) in iteration 1, then SK(i)>SK(j) for all future iterations. Obviously, the initial coloring of nodes is not permanent, as it is not set within the algorithm itself.)

Once the structural characteristic coloring has been found, we can easily label the network in question by assigning node numbers in descending color order. If the graph still contains some identically-colored nodes, their specific ordering is irrelevant (so long as they are properly ordered with respect to all differently-colored nodes). In many cases, such non-degenerate color sets are due to structural equivalence (Lorrain and White, 1971); hence, their orderings will not affect the Hamming metric¹⁸. It is possible, however, for algorithmic failures to cause non-unique labelings of some non-equivalent nodes¹⁹. The degree to which this affects assessments of Hamming distance can vary, but (as with any heuristic method) caution is advised.

Illustrative Application of the Color-Splitting Algorithm

Figure 5 presents a simple directed graph. Given that the graph is uncolored (or 1-colored), how would the foregoing algorithm determine a unique labeling?

<Insert: Figure 5. A 1-Colored Graph with Arbitrary Labels>

To determine this, let us "run through" the instructions and observe the results. The intermediate results are also shown in Figure 6. Initially, our G consists of the nodes A-I (3), d is

¹⁸ Structurally equivalent actors have identical relations with identical (in this case, identically-colored) alters. Thus, their ordering cannot change the value of any entry in a sociomatrix so long as they are A) part of a coherent block which B) is in the same position vis a vis all other nodes.

¹⁹ Informal observation seems to indicate that the algorithm is particularly vulnerable to regular equivalence.

equal to 0 (1), and v is equal to 0 (2). Proceeding to step (4), we note that: $SR_A(0,0)=3$; $SR_B(0,0)=1$; $SR_C(0,0)=0$; $SR_D(0,0)=2$; $SR_E(0,0)=1$; $SR_F(0,0)=1$; $SR_O(0,0)=1$; $SR_H(0,0)=0$; $SR_H(0,0)=3$;. Thus, we can already split G into $\{A,I\}$, $\{D\}$, $\{B,E,F,G\}$, and $\{C,H\}$.

In (5), we now attempt to split the identically-colored subsets, observing that: $SC_A(0,0)=2$; $SC_B(0,0)=2$; $SC_B(0,0)=1$; $SC_B(0,0)=1$; $SC_B(0,0)=0$; $SC_B(0,0)=0$; $SC_B(0,0)=0$; and $SC_B(0,0)=0$; and $SC_B(0,0)=0$. Applying the same ordering rule as we used in (4), we are able to arrive at the division {A,I}, {D}, {G}, {B,E}, {F}, {H}, {C}.

At this point (6), we note that d < N-1 and our ordering is not degenerate, but $v=2^d-1$; thus, we fail the condition of (6) and proceed to (7), where we let d=1 and return the two sets $G_1=\{A,I\}$ and $G_2=\{B,E\}$ to step (2) (set v=0).

<Insert: Figure 6. A Sample Color-Splitting Process>

For (4) G_1 , we can see that: $SR_A(1,0)=8$; and $SR_I(1,0)=7$. This splits G_1 into $\{A\}$, $\{I\}$. For (4) G_2 , we find that: $SR_B(1,0)=4$; and $SR_E(1,0)=3$. This final division splits G_2 into $\{B\}$, $\{E\}$.

At this point, the ordering {A}, {I}, {D}, {G}, {B}, {E}, {F}, {H}, {C} is degenerate, and we are finished. The descending-order labeling which results from this process can be seen in Figure 7.

<Insert: Figure 7. Sample Graph After Labeling>

The Special Case of Structural Equivalence

Structural equivalence, as has been noted, poses a special problem for the Recursive Color-Splitting Algorithm: because structurally equivalent actors have the same relations with the same alters, the structural row and column characteristics of those actors must be the same for any depth. Because of this, the RCS algorithm will be unable to establish a non-arbitrary labeling of structurally equivalent nodes; further, the algorithm will have to traverse the entire set of walks before discovering this. While the latter is an issue of some concern in implementation, the former is not, as it turns out, a problem for our analysis. Due to the aforementioned definition of structural equivalence, rows and columns of equivalent actors must be identical $(A_{ik}=A_{jk}, A_{ki}=A_{kj} \forall k\in A)$. If this is the case, then the order of structurally equivalent actors cannot affect the Hamming distance between two matrices: since the actors are structurally *identical*, no "flips" of matrix entries are needed to map one onto the other. The fact that the algorithm cannot discriminate between these actors, then, is of no consequence.

It should be pointed out that the same cannot be said of other sorts of equivalence, which may or may not affect the RCS algorithm (depending on the maximum depth of recursion, the structural properties of the network, etc.). Regularly equivalent actors, for instance, may in some cases be reordered in ways which do not minimize Hamming distance. (An example of this may be seen in Figure 8.) More work is required to identify the specific network features which cause RCS (or the other algorithms here considered) to fail in producing canonical labelings.

<Insert: Figure 8. An Example of Orderings Produced Under Regular Equivalence>

The Nodal Degree Ordering Algorithm

As has been indicated, we are here interested in algorithms which will produce a unique labeling of nodes given some initial coloring. As this labeling is to be canonical, it must exploit particular features of the individual graph which is the subject of the labeling process. One category of features which suggests itself for such a process is the degree centrality of nodes within the digraph; considered separately as indegree and outdegree, this measure is equivalent to the column or row sums (respectively) of a particular node. Our second method, then, for labeling graphs is an ordering based on nodal degree alone, without concern for path structure (unlike the RCS algorithm). As we shall see, this omission limits the ability of the nodal degree ordering to exploit subtle differences between graphs, but greatly simplifies it computationally20.

To utilize nodal degree in the labeling process, we may begin simply by sorting nodes within color groups. While this may be implemented in a number of ways, we shall here consider only ascending orderings sorted primarily on indegree and secondarily on outdegree. That is, we say that given

[15]
$$R_a = \sum_{i=1}^{N} \mathbf{A}_{a,i}, \quad R_b = \sum_{i=1}^{N} \mathbf{A}_{b,i}$$

[16] $C_a = \sum_{i=1}^{N} \mathbf{A}_{i,a}, \quad C_b = \sum_{i=1}^{N} \mathbf{A}_{i,b}$

[16]
$$C_a = \sum_{i=1}^{N} \mathbf{A}_{i,a}, \quad C_b = \sum_{i=1}^{N} \mathbf{A}_{i,b}$$

the ordered coloring k is defined on the nodes a and b according to the following algorithm:

1). If
$$R_a > R_b$$
, $k(a) > k(b)$; if $R_a < R_b$, $k(a) < k(b)$; else (2)

2). If
$$C_a > C_b$$
, $k(a) > k(b)$; if $C_a < C_b$, $k(a) < k(b)$; else (3)

3). k(a) = k(b)

This degree-based labeling has certain practical advantages. First and foremost, it is computationally facile, both in terms of difficulty of implementation and in speed of execution. Second, both indegree and outdegree are well defined for all manner of networks (including

²⁰ In fact, the nodal degree ordering corresponds to a degenerate case of RCS in which only structural characteristics

continuous or multiply valued networks), and their distributions can be calculated mathematically given some basic assumptions. Unfortunately, however, the technique has some drawbacks as well. Highly structured networks (such as organizational charts) often exhibit very low variance on indegree and outdegree, thus limiting the discriminatory power of this measure. Further, because the nodal degree labeling considers only a local property of nodes, it is unable to identify structural differences that manifest only in non-local patterns (e.g., regularly equivalent nodes which are not structurally equivalent.). For these reasons, we should expect the nodal degree method to be of little use when comparing highly structured, self-similar networks, but to be reasonably effective when comparing highly diverse networks (e.g., weakly constrained random graphs).

The Weighted Degree Ordering Algorithm

The nodal degree ordering algorithm, above, uses the structural property of nodal degree to effect a labeling, but does not discriminate between degrees save via raw magnitude. RCS, as an extension of this notion, also gives weight to walks throughout the graph, but is relatively slow. Given the apparent desirability of drawing on network properties other than raw degree, is there a simple weighting system which might be used to effect an ordering of nodes? One precedent for such an approach can be taken from previous work in this area. In his 1996 paper, "An Algorithm to Generate Connected Graphs," John Skvoretz draws upon an algorithm of Read (1978) to establish a canonical labeling of directed graphs (Skvoretz, 1996) which maps isomorphism classes onto unique representations. Simply stated, this algorithm attempts to

of distance 0 are considered.

identify the permutation of a dichotomous sociomatrix which maximizes its value when expressed as a binary number. More particularly, if we define the value function V of a permutation matrix \mathbf{P} such that

[17]
$$V(\mathbf{P}) = \sum_{i=1}^{N} \sum_{j=1}^{N} (\mathbf{AP})_{ij} 2^{(N(j-1)+N-i)}$$

then we attempt to find that **P** for which V(**P**) is maximized. Under this representation, then, the problem of canonical labeling is mapped onto a problem of lexicographic sorting (Cai et al, 1992). In his paper, Skvoretz proposes using an exhaustive search to identify the proposed canonical labelings of all (small) connected graphs. Unfortunately, however, this process is of order N!, and is hence not feasible for networks of significant size. While various alternative methods may be used to search the space of permutations for one which maximizes the proposed value function, we here will consider a very simple heuristic which is closely related to the nodal degree ordering. Given

[18]
$$VR_a = \sum_{i=1}^{N} \mathbf{A}_{a,i} 2^{(N-i)}, VR_b = \sum_{i=1}^{N} \mathbf{A}_{b,i} 2^{(N-i)}$$

[19]
$$VC_a = \sum_{i=1}^{N} \mathbf{A}_{i,a} 2^{(N-i)}, VC_b = \sum_{i=1}^{N} \mathbf{A}_{i,b} 2^{(N-i)}$$

we may define the ordered coloring k on the nodes a and b according to the following algorithm:

1). If
$$VR_a > VR_b$$
, $k(a) > k(b)$; if $VR_a < VR_b$, $k(a) < k(b)$; else (2)

2). If
$$VC_a > VC_b$$
, $k(a) > k(b)$; if $VC_a < VC_b$, $k(a) < k(b)$; else (3)

3).
$$k(a) = k(b)$$

This heuristic, then, is not an implementation of Read's lexicographic sorting, nor is it intended as such; instead, it is an attempt at using the binary value representation as a mechanism for weighting the indegrees and outdegrees of individual nodes. The results reported here should thus be interpreted as following from the particular weighting algorithm rather than the previous labeling approach. Nevertheless, the two are alike in at least two respects: both attempt to concentrate the 1's of the sociomatrix in a particular location within the matrix, and both are quite sensitive to relatively small differences in graph structure.

The Random Permutation Algorithm

While we may in many cases have a best-case criterion against which to judge algorithmic performance, it is also critical that we have a reasonable worst-case estimate. One such estimate may be achieved via the random permutation algorithm: a process that randomly allocates labels to nodes within initially specified color groups. While a random selection from the space of permutations is unlikely to produce a labeling which actually *maximizes* the hamming distance between any two matrices (the true worst-case), it provides a useful base-line performance level by avoiding any optimization of node labeling whatsoever. It is expected that all three of the alternative algorithms presented will outperform the random permutation algorithm on all effectiveness metrics.

Comparison of the Labeling Algorithms

All of the proposed canonical labeling algorithms are generally effective at producing unique labelings of N-colored graphs. However, as we noted earlier, if the goal is to compare and contrast graphs it is not sufficient to merely label the nodes. Rather, the labelings that are produced must satisfy (or at least approximate) our criterion for canonicality with respect to the structural distance. This is especially true for cases in which the graphs to be compared are known to differ in some way; ideally, there will be a strong linear relationship between this underlying structural difference and the observed Hamming distance between the two networks. (Indeed, optimally the two quantities should be equal after labeling.)

To effect a preliminary test of the validity of the three labeling algorithms for this specialized purpose, we ran a series of three tests. First, we performed a virtual experiment in which we generated a large number of graphs which were more or less typical of what might be expected for social structures with a small to moderate number of nodes. Each of these graphs was then copied and "tweaked": that is, some number of ties (chosen at random) were "flipped", so as to produce a slight, known difference between the two networks. After being modified, the copied graphs were randomly permuted, and the three labeling algorithms (along with the random permutation algorithm) were executed on each. Once the graphs were labeled, the Hamming metric was used to find the observed distance between the two graphs (the original and the permuted tweaked graph); this distance was then compared to the criterion (tweak) distance and to the distance produced by the random permutation algorithm. Our second assessment, unlike the first, focused on isomorphic graphs which were randomly prelabeled. For each of the conditions from the first virtual experiment (excepting the omission of the tweak treatment), two identical copies of a random graph were given arbitrary labels, the labeling algorithms were applied to each, and the observed Hamming distance between them was measured. This distance

was then compared to the ideal distance of zero for all cases, in order to assess the effectiveness of the labeling algorithms at mapping all members of an isomorphism class onto a single labeled graph. Our third, and final, test of the labeling algorithms examined the ability of the algorithms to produce maximal clustering (i.e., minimal mean distance) of central graphs under varying conditions. For this assessment, monte carlo methods were used to produce a variety of graphs with particular properties; in each case, these graph sets were labeled and their central graphs identified. The distribution of distances of matrices from the central graph was then examined in order to compare the degree to which algorithms were able to induce clustering.

In all of the above tests, our primary interest is in comparing the relative effectiveness of these algorithms to a random labeling of nodes. In two of the tests, we are also able to directly and systematically evaluate algorithm effectiveness against a known performance criterion (the minimal Hamming distance); in one of these, we know the criterion precisely and without error. There are, of course, a variety of other assessments that one might seek to perform in evaluating these labeling algorithms. For instance, one might focus on networks with particular properties (e.g., those found in cognitive or ego networks), or to evaluate central graph identification against a more precise criterion. Furthermore, there are other potential labeling methods to those described here, each of which having its own advantages and disadvantages²¹. Our analysis should be viewed, then, as preliminary; future research into alternative methods and standards of evaluation is encouraged.

Random Graph Comparison

Our first comparison of the labeling algorithms involves our most basic functional requirement: the ability to minimize the labeling distance between matrices. To investigate this area of performance, we employed a virtual experiment involving a large number of random graphs produced under various constraints matching those typically found in social network research. In each case, the graphs were created, copied, and "tweaked": that is, one of the copies was altered by randomly flipping $(0\rightarrow 1,1\rightarrow 0)$ a prespecified number of ties. Each copy was then randomly permuted and individually labeled using one of the three coloring algorithms (and the random coloring algorithm). At this point, the Hamming distance between the newly labeled copy and the re-labeled initial graph was found, and compared to the criterion (the number of flipped ties) and to the baseline (the Hamming distance of the random permutation algorithm). By considering the resultant distribution of Hamming distances across a range of conditions, then, we hope to be able to assess the effectiveness of the three proposed labeling heuristics at aligning graphs for maximal similarity.

<Insert: Table 1: Conditions for the Virtual Experiment>

The conditions for the virtual experiment are given in Table 1. As can be seen, graph size and structure were varied along several dimensions, with a total number of 420 different conditions. Further, six different levels of tweaking were examined for each structural condition, and 25 matrices were independently drawn from each of these cases, resulting in a total of 63,000 observations on 2,520 conditions per labeling method (the random baseline was computed for each other method's sample, and on its own; RCS was limited to a search depth of five hops for computational reasons). Due to natural limitations, some densities precluded the examination of

²¹ Of particular long-term interest might be dynamic methods which use genetic algorithms, simulated annealing, or other approximate optimization methods to search the space of permutations for optimal labelings.

some reciprocity values²²; in these cases, the closest achievable value was substituted. As these represent regions of the parameter space for which no observations can exist, these "omissions" should not substantially affect our findings.

<Insert: Figure 9. Boxplots of Hamming Experiment Data>

Preliminary examination of experimental data for all conditions indicated extreme skew and generally poorly behaved distributions on most distance variables (see Figure 9 above). To counteract this, a logarithmic transformation was applied to all variables prior to regression. Out of those transformations tried, the natural logarithm appeared to induce the best fit (as measured both by R² and by examination of residuals) in the final models; this transformation also had a natural interpretation in this case – that of the power law – which was sensible in light of the underlying phenomenon of interest. Examination of regressor variables for multicollinearity found no serious problems, as expected due to the nature of the experimental design.

<Insert: Table 2: Prediction of Structural Distance from Observed Hamming Distance>

In order to assess the effectiveness of the proposed canonical labeling algorithms, we seek to relate the observed Hamming distance to the "real", underlying structural distance. Ideally, these two would be the same, or nearly so; in fact, however, we expect that imperfections in the labeling algorithms will prevent the labeling distance (D_L) from taking its minimal value (0) and that the observed Hamming distance will thus be increased by a certain amount. How, then, can we characterize algorithmic performance? One, very simple method which we might employ to gain a general idea of the way in which the algorithms behave is a univariate regression of structural distance on the observed distance after labeling. While such a regression has little explanatory value (it cannot, for instance, tell us why a particular algorithm performs in a given

²² For instance, a density of 0.8 precludes a reciprocity of 0.0.

fashion), it can give us an overall impression of the degree to which a direct, "naïve" application of the labeling algorithms is likely to overstate the actual Hamming distance between graphs.

The most basic results from this analysis of the virtual experiment are given in Table 2, which lists the coefficients, R²s, residual standard errors, and correlation coefficients for the four regressions. Due to the fact that these regressions are intended primarily to model the effect (in terms of degree of overstatement of structural distance) of direct use of the labeling algorithms, all models in Table 2 are constrained to pass through the origin. Although a rough analysis, such models are valuable because they provide a prediction of raw performance in the absence of additional information - a worst-case scenario. As can be seen from the included coefficients, the RCS algorithm is most effective at producing Hamming distances close to those expected from an ideal model (in which case the coefficient would be approximately 1.0²³), and has the highest overall R², lowest residual standard error, and highest correlation coefficient. The nodal degree ordering is the next most effective algorithm, followed by the weighted degree ordering; all, however, outperform the random permutation algorithm both in terms of magnitude of coefficient and in terms of variance explained. That said, the differences between algorithms are not large (although all differences between residual standard errors are significant), and none of those we consider approach optimal performance.

<Insert: Table 3: Prediction of Hamming Distance by Condition>

While these results are suggestive, they are admittedly imprecise: they tell us nothing regarding the sensitivity of the labeling algorithms to various experimental conditions.

Information of this sort can be found in Table 3, which presents a series of regressions of observed Hamming distance on various experimental parameters using the following model:

 $[20] \log (D_o + 1) = \beta_0 + \beta_1 \log (N+1) + \beta_2 \log (Den+1) + \beta_3 \log (\operatorname{Re} \operatorname{cip} + 1) + \beta_4 \log (D_S + 1) + \beta_5 \log (\operatorname{Color} + 1) + \beta_5 \log (\operatorname{Col$

(where ε is assumed to be approximately normally distributed). Note that this model corresponds to the following nonlinear equation:

[21]
$$D_o = e^{\beta_0} (N+1)^{\beta_1} (Den+1)^{\beta_2} (\operatorname{Re} cip+1)^{\beta_1} (D_S+1)^{\beta_4} (Color+1)^{\beta_5} e^{s} -1$$

Thus, negative coefficients signify tendencies towards lower Do values, but do not imply Hamming distances which are themselves negative! As before, we should expect the structural distance coefficient to approach 1.0 as effectiveness increases; we should bear in mind, however that we are here actually postdicting observed Hamming distance from the structural distance rather than the other way around. The qualitative influence of various factors on algorithm performance can be assessed based on the information in Table 3. In particular, we find that all algorithms see an increase in observed Hamming distance as a function of N (roughly on the order of N2): this may reflect the fact that the error cost of suboptimal orderings can be expected to be on the order of the number of nodes24, and that the number of potential ties to be flipped is equal to N2. Likewise, increasing density appears to increase observed Hamming distance, while increasing reciprocity decreases it very mildly. Although this apparent favoritism for highly symmetric, sparse matrices is not fully obvious, it most likely results at least in part from the natural distribution of Hamming distances²⁵. The fact that these results are present in the random permutation case further supports this line of reasoning. The finding that two-colored graphs result in lower Hamming distances than their one-colored counterparts across methods is

²³ In practice, some deviation from 1.0 will be observed due to the randomized implementation of tweak, and from the small but nonzero probability that a series of tweaks will map a graph onto one of its automorphisms.

This can be seen from a worst-case scenario in which an all-0 node and an all-1 node are switched: the added

Hamming distance will be approximately 4N-2.

likewise suggestive: as two-colored graphs have considerably smaller permutation spaces than one-colored graphs, we would expect the nature of the search task to be inherently easier for these cases.

<Insert: Table 4: Diagnostic Error Regression>

Another means of addressing the question of what determines algorithmic behavior is an error regression; that is, a model which attempts to predict the absolute difference between the structural distance and the observed Hamming distance from experimental conditions. Such a model has been fitted in Table 4 above:

[22] $\log(D_0 - D_s|+1) = \beta_0 + \beta_1 \log(N+1) + \beta_2 \log(Den+1) + \beta_3 \log(Re \, cip+1) + \beta_4 \log(Color+1) + \varepsilon$ As the Table reveals, the results of this regression are quite similar to the previous one. Graph size and density have strong positive effects, while reciprocity and two-coloring are more weakly negative on all models. As above, we find that those methods whose performance is the most mediocre are those whose behaviors are most easily predicted; across all cases, however, goodness-of-fit is reasonably good. This last suggests that it may be possible to use these regressions in conjunction with the algorithms themselves to arrive at more effective estimates of the structural distance generally; while this is an idea with a great deal of promise, it is beyond the scope of this paper.

<Insert: Table 5: T-Tests of Adjusted Errors Across Methods>

Having examined the relationship between our experimental conditions and the observed Hamming distance, we now seek to bolster our analysis by investigating the error rates produced by the various labeling methods. In particular, we wish to verify here that the labeling heuristics

²⁵ The distribution of Hamming distances between any two matrices is not normal, and varies greatly with size and density. While this lessens the power of our regression, we employ it here as a first step, in the absence either of

are significantly better than random labelings on identical samples26. Likewise, we would like to make comparisons of error rates across samples, in order to compare the proposed canonical labeling algorithms with each other. Table 5 explores this question by presenting a series of ttests of the errors produced by labeling algorithms versus the errors produced by random label assignment and versus the other methods, across all conditions. Note that, for purposes of this test, we have adjusted the errors produced on each observation (as used in the above regression) by dividing them by the square of the graph size at that observation. The rationale for this is clear: since we have seen that the magnitude of error scales in N2, normalizing the errors by this quantity prevents large graphs from dominating the sample. As it happens, this transformation is somewhat superfluous; the results reported in Table 5 were typical of those found using raw data, log transformed data, etc.27 As can be seen from Table 5, all three methods produced significantly fewer errors than the random permutation algorithm -- this reinforces our earlier finding that prediction of structural distance from observed Hamming distance was more accurate when labeling heuristics were employed. The proposed labeling algorithms were also significantly different from each other, with RCS outperforming nodal degree ordering, and both surpassing the weighted degree ordering. Thus, our earlier impressions as to the efficacy of the labeling algorithms are confirmed by a direct comparison of absolute errors under identical conditions.

Isomorphic Structures

better tools or of detailed distributional information.

²⁶ As opposed to equivalent samples, as were considered previously.

between algorithmic performance and search depth; both because of the computational expense of high-depth RCS labelings and because of the fundamental relationship between the nodal degree ordering and RCS, these issues were felt to be important ones.

<Insert: Figure 10. Order Statistics of Observed Hamming Distance for Isomorphic Graph Experiment>

The order statistics of the observed Hamming distance, by method, are presented in Figure 10. As expected, the random permutation algorithm generally did not minimize the Hamming distance between matrices, as evidenced by its high levels of error throughout the distance distribution. The weighted degree ordering, also as expected, exhibited noticeably and consistently superior performance to the random labeling, but fell substantially behind the other algorithms. Herein lies the most striking finding: dramatic improvements in labeling were had by employing the degree ordering and RCS algorithms, and RCS error was negligible for all but the worst cases as early as depth 2! This finding is confirmed by the values in Table 6 below. Even at the initial stages of the RCS process, the degree of error in classification falls off quickly, and substantial errors are restricted to the fourth quartile of the total distribution by the second order recursion.

<Insert: Table 6: Observed Hamming Distances for Isomorphism Experiment>

In addition to calculating the error in specification of the Hamming distance, it is also possible to simply count the raw number of occasions in which the observed Hamming distance is positive. Such events are instances of *labeling failure*, in which the canonical labeling algorithm has failed to identify a labeling such that the Hamming distance is minimized.

differed significantly from the random baseline under all transformations attempted, however.

Because labeling failures may involve a wide range of error magnitudes, then, they provide an alternative measure of algorithmic effectiveness.

<Insert: Figure 11. Total Number of Labeling Failures for Isomorphic Graph Experiment>

Observe the failure rates for Degree and RCS algorithms depicted in Figure 11. As with the distribution of Hamming distances, we find that the failure rate falls off sharply with increasing search depth, with marginal gains diminishing quickly after a depth of two hops. More interestingly, we find a clear nonlinear pattern in the relationship between graph size and labeling failure: initially, increasing size appears to increase the failure rate, which then begins to peak and fall off slowly after passing its maximum (generally between 10 and 20 nodes). The "peak value" of the failure rate seems to depend on the search depth; deeper searches lead to early, small peaks and more rapid falloff, while shallow searches have longer and steeper ascents. The nodal degree ordering algorithm (corresponding to RCS at depth 0) is particularly notable in this respect. From our sample, it is not obvious that the failure rate of the degree ordering algorithm falls off in the limit of large size; we should expect, however, that this will occur for theoretical reasons. Given the shape of the curve in question, then, a reasonable prediction would anticipate a gradual falloff of the degree ordering failure rate, beginning perhaps by the size=50 mark. Such a prediction may be tested in future research. Specific values for failure counts by graph size and search depth are listed in Table 7, below.

<Insert: Table 7. Number of Failures under RCS (and Degree)>

The implications of these findings are quite significant, particularly in light of the previous experiment. Despite the fact that none of the algorithms considered – including RCS – were highly effective at minimizing the labeling distance for the general case, it is apparent that RCS at the least does map members of isomorphism classes onto unique representations with a high

degree of reliability. Given this apparent disparity, then, we must conclude that the problem of identifying isomorphism classes is actually a very different one from the general distance minimization problem, and that the class of algorithms which can satisfy the former problem may not typically be terribly useful with respect to the latter²⁸. This in turn suggests that techniques which have been developed to conduct isomorphism testing (e.g. (Luks, 1982), (Babai and Luks, 1983), (Kučera, 1987)) may prove ineffectual at facilitating comparison of unlabeled graphs in the general case. On the other hand, the success of the RCS algorithm at matching isomorphic graphs without extensive searching is quite promising. For short maximum depths, RCS is not terribly expensive, and it would seem that (for this sample, in any event) only a relatively small number of walks must usually be considered in order to classify vertices effectively.

Central Graphs

As noted, the central graph (Banks & Carley, 1994) is that graph which contains those edges in 50% or more of the set of graphs on which it is calculate. It is also analogous to the mean for variable level data, in that it can be considered an estimator which minimizes the sum of squared Hamming distances between graphs within the set on which it is defined. After labeling the nodes for the sample graphs given in Figure 2, for example, we calculated the central graph. This central graph is shown in Figure 12. For the four original graphs, their Hamming distance from this graph is (clockwise in Figure 2) 0, 8,4, and 4. In this case, this gives us the minimal set of Hamming distances of these graphs from the central graph – the choice of labeling has allowed us to minimize D_L for comparisons within this set.

²⁸ Although the latter algorithms must, in the ideal case, be capable of satisfying the former problem.

<Insert: Figure 12. Central Graph for Four Sample Graphs>

This example serves to indicate our more general goal: here, we are interested in investigating the properties of the three proposed labeling heuristics with regards to the identification of central graphs in a variety of contexts. To accomplish this goal, we here consider a large number of randomly generated graphs. In each case, by examining the degree to which the algorithms are able to induce clustering (thereby minimizing the mean distance from the central graph) we are able to evaluate the properties of each.

In order to effect the test described above, we considered a large number of randomly generated graphs whose properties were varied systematically. For each condition, we labeled the total set of matrices using each of the above labeling algorithms, and then found the set's central graph. Once the central graph was identified, it became possible to find the Hamming distance of each matrix in the set from the set's central graph; the distribution of distances thus generated was then analyzed in order to evaluate algorithmic performance.

Table 8 shows the set of conditions for the central graph experiment. Three different graph sizes and three density settings were specified, for a total of nine conditions (one of which was omitted due to computational constraints). For each condition, 100 matrices were randomly produced, for a total of 800 matrices. The matrices within each condition were labeled and stacked, and the central graph analysis was applied to these stacked matrices; this avoided possible confounding effects from mixing matrices across conditions. The distribution of Hamming distances was analyzed both within condition and across all conditions, although the results which follow consider only the latter. The within condition results were consistent with

the aggregate results for all conditions, although some results did not reach significance in the former case.

<Insert: Table 8: Conditions for the Central Graph Experiment>

The aggregate results for the central graph analysis are given in Table 9. As can be seen, the same basic performance pattern observed elsewhere holds in this case, with nodal degree ordering and RCS significantly outperforming the weighted degree ordering and the random permutation algorithm. The lower mean Hamming distance and tighter dispersion of the three labeling algorithms demonstrates that they do indeed induce clustering and facilitate the identification of central graphs. This performance characteristic can be expected to benefit other methodologies (e.g., consensus structures, mean graphs) which attempt to identify common structural characteristics of matrix sets as well.

<Insert: Table 9: Central Graph Analysis - Cumulative Results>

Discussion

In this paper, we have attempted to take an initial approach to the problem of identifying central features of unlabeled graph sets. We have defined a mathematical framework which draws upon the notion of the Hamming distance to construct theoretically meaningful measures of structural distance, and have shown that there exists a particular type of canonical labeling algorithm which permits the direct inference of structural distance from the Hamming distance. Following this, we have considered several potential canonical labeling algorithms, and we have tested the effectiveness of these algorithms at minimizing the labeling distance between directed graphs under various conditions. Although we have found fairly consistent results across tests, a

number of questions remain. In particular, computational limitations forced us to examine only a relatively small subset of the set of possible graphs: very large and very dense graphs, for example, were not considered. Likewise, in sampling across dimensions such as size, density, and reciprocity, we have not performed a detailed examination of algorithmic performance within categories; since some social structures (e.g., authority relations, familial ties) have very specific properties, it would be useful to have more complete information on algorithmic behavior for these "special cases".

Another area in which more work is clearly needed is in understanding the effects of preliminary coloring on algorithmic performance, and on Hamming distance generally. Although non-degenerately colored graphs are not the norm in social research, there are excellent theoretical reasons for employing them; our methods, alas, have thusfar barely begun to address this class of problems. As more is learned regarding the effects of coloring on network variables, and as more statistical tools become available to deal with this issue, it is anticipated that this means of representing social structure will become more widely used. This paper, hopefully, adds to this literature by presenting a general means by which unlabeled graphs may be transformed for use by conventional, labeled methods.

Insofar as we deal with specific algorithms, a number of issues are raised regarding both effectiveness and efficiency. While we found that, overall the RCS algorithm was most effective at minimizing Hamming distance between sociomatrices (particularly in the case of isomorphic graphs) and inducing clustering in central graph analysis, we also found the method to be extremely computationally expensive. While various means are available of improving

performance29, it is notable that the nodal degree ordering algorithm (a degenerate form of RCS) was nearly as effective over a range of conditions, and at dramatically reduced cost. This suggests that, at least for relatively heterogeneous populations of networks, degree may be preferable to the more thorough - but slower - RCS method. On the other hand, our failure analysis of isomorphic graphs appears to indicate that (for random graphs, in any event) a search depth of only two hops is sufficient to gain most of the performance advantages of RCS. As this is not terribly costly, such an abbreviated RCS process may be an effective compromise for a wide variety of circumstances. The weighted degree ordering algorithm provides yet another alternative, but was not found to perform as well as the other methods in our study. This may be due to the manner in which the heuristic was implemented: it is possible that a more sophisticated weighting system would yield increased effectiveness. While more work should be done to consider when and where the a simple weighted degree heuristic should be used, it seems important to note that, by placing disproportionate weight on certain portions of the network, such algorithms may be expected to have difficulty with noisy data. Given our findings vis a vis isomorphism testing and more general canonical labeling, it is suspected that this is a serious flaw with the weighting approach as we have here conceptualized it.

In addition to considering these questions relating to dyadic comparison, we have considered the relationship between our proposed canonical labeling algorithms and the central graph. As we have indicated, a prudent choice of labels permits the inference of central structures which reflect the underlying distribution of structural distances between graphs rather than the surface distances (which are influenced by arbitrary choice of node labels). Empirical tests using our proposed labeling algorithms confirm this intuition, indicating significantly lower distances to

Such as identifying and flagging structurally equivalent nodes prior to labeling.

the central graph after labeling than previous to it. Further work in this area should consider other extensions of this type: application of the results of this paper to algorithms for classifying populations of networks via clustering methods; generalization of the central graph to other graphical estimators; and extension of the distance metric to permit the definition of measures on the space of graphs, are all potential areas of development which could benefit from access to effective canonical labeling routines.

As a final, theoretical note, it is important to emphasize that the priority given here to the elimination of distance effects due to labeling does not imply that labeling distance is simply "noise" to be removed in all circumstances. Understanding the ways in which labels affect graph comparison may result in advances on a number of fronts, not the least of which being the examination of meaningfully prelabeled structures. Development of the Hamming decomposition, by the same token, is a possible avenue for future research. The distribution of the labeling distance, for instance, is clearly related to graph automorphisms, and likewise to methods such as the Quadratic Assignment Procedure (Krackhardt, 1988) which explore the space of permutations in order to compare similarities. The importance of labels for inference should not be underestimated; at the same time, however, we must endeavor to choose our procedures – and our labels – with care so as not to ask the wrong questions of our data sets.

Conclusion

We have considered here three heuristic algorithms for labeling unlabeled or colored graphs: a recursive color-splitting algorithm, a nodal degree ordering algorithm, and a weighted degree ordering heuristic. Labeling graphs implicitly aligns their structures, mapping graphs onto canonical representations of their isomorphism classes. Once the graphs are labeled in this fashion, similarities and differences can be measured and the central graph calculated using standard methodologies. Given the central graph, it is then possible to generate other features of the distribution of structural distances and to statistically evaluate differences and similarities between graphs. Such techniques are useful in determining differences in those aspects of social structures that can be represented as directed graphs, and can similarly be used to identify commonalties between social structures across a wide range of settings. Ultimately, it is hoped that the basic framework and tools developed here will further the creation of a unified set of statistical methods for the representation, characterization, and comparison of structural data.

References

[Aldrich & Mueller, 1982] H.E. Aldrich and S. Mueller, "The Evolution of Organizational Forms: Technology, Coordination, and Control." In *Research in Organizational Behavior*, vol. 4, edited by B.M. Staw & L.L. Cummings. Greenwich, CN: JAI, 33-87, 1982.

[Babai & Luks, 1983] L. Babai and E. Luks, "Canonical Labeling of Graphs," Proceedings of the 15th ACM Symposium on the Theory of Computing, 1983.

[Banks & Carley, 1994] D. Banks and K. Carley, "Metric Inference for Social Networks," Journal of Classification, 11, 121-149, 1994. [Cai et al., 1992] J. Cai, M. Fürer, and N. Immerman, "An Optimal Lower Bound on the Number of Variables for Graph Identification," Combinatorica, 12(4), 389-410, 1992.

[Carley, 1995] K.M. Carley, "Automatic Restructuring of the C2 Structure and Performance," Proceedings of the 1995 International Symposium on Command and Control Research and Technology. June. Washington D.C., 1995.

[Carley, 1986] K.M. Carley, "An Approach for Relating Social Structure to Cognitive Structure," *Journal of Mathematical Sociology*, 12, 137-189, 1986.

[Carley, 1984] K.M. Carley, Constructing Consensus. Unpublished doctoral dissertation, Harvard, 1984.

[Fligstein, 1985] N. Fligstein, "The spread of the multi-divisional form among large firms, 1919-1979," American Sociological Review, 50, 377-391, 1985.

[Hamming, 1950] Hamming, "Error Detecting and Error Correcting Codes," Bell System Technical Journal, 29, 147-160, 1950.

[Hannan & Freeman, 1989] M.T. Hannan, and J. Freeman, Organizational Ecology. Cambridge, MA: Ballinger, 1989. [Krackhardt, 1987] D. Krackhardt, "Cognitive Social Structures," Social Networks, 9, 109-134, 1987.

[Krackhardt, 1988] D. Krackhardt, "Predicting with Networks: Nonparametric Multiple Regression Analysis of Dyadic Data," Social Networks, 10, 359-381, 1988.
[Krackhardt, 1994] D. Krackhardt, "Graph Theoretical Dimensions of Informal Organizations."

In Computational Organization Theory, edited by K.M. Carley and M.J. Prietula. Hillsdale,

NJ: Lawrence Erlbaum Associates, 1994.

[Kučera, 1987] L. Kučera, "Canonical Labeling of Regular Graphs in Linear Average Time," Proceedings of the 28th Annual Symposium on the Foundations of Computer Science, Los Angeles, CA, 12-14 Oct., 1987

[Lim, 1989] C. Lim, "Canonical Transformations and Graph Theory," Physics Letters (A), 138(6-7), 258-266, 1989.

[Lorrain & White, 1971] F. Lorrain, and H. C. White, "Structural Equivalence of Individuals in Social Networks," *Journal of Mathematical Sociology* 1: 49-80, 1971

[Luks, 1982] E. Luks, "Isomorphism of Graphs of Bounded Valence Can Be Tested in Polynomial Time," Journal of Computer and Systems Sciences, 12, 42-65, 1982. [McKelvey, 1982] B. McKelvey, Organizational Systematics: Taxonomy, Evolution, Classification. Berkeley, CA: University of California Press, 1982

[Pospichal & Kvasnička, 1996] J. Pospichal and V. Kvasnička, "Pruning the Search Tree in the Constructive Enumeration of Molecular Graphs," *Discrete Applied Mathematics*, 67, 189-207, 1996.

[Read, 1978] R. C. Read, "Every One a Winner or How to Avoid Isomorphism Search When Cataloguing Combinatorial Configurations," Annals of Discrete Mathematics, 2, 107-120, 1978.

[Romanelli, 1989] E. Romanelli, "Environments and Strategies of Organizational Start-up: Effects on Early Survival," Administrative Science Quarterly, 34, 369-387, 1989.

[Romney et al., 1986] A.K.Romney , S.C. Weller and W.H. Batchelder, "Culture as consensus: A theory of culture and informant accuracy," *American Anthropologist*, 88(2), 313-338, 1986.

[Skvoretz, 1996] J. Skvoretz, "An Algorithm to Generate Connected Graphs," Current Research in Social Psychology, 1(5), 1996.

[Wasserman & Faust, 1994] S. Wasserman, and K. Faust, Social Network Analysis: Methods and Applications. Cambridge University Press, 1994.

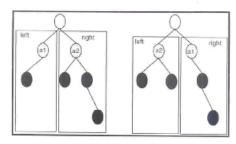


Figure 3. Two Colored Networks

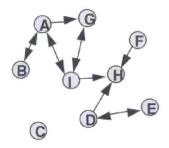


Figure 5. A 1-Colored Graph with Arbitrary Labels

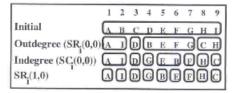


Figure 6. A Sample Color-Splitting Process

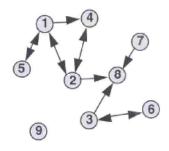


Figure 7. Sample Graph After Labeling

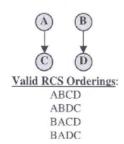


Figure 8. An Example of Orderings Produced Under Regular Equivalence

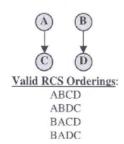


Figure 8. An Example of Orderings Produced Under Regular Equivalence

Table 1: Conditions for the Virtual Experi	ment
--	------

	you the state isoperiment
Treatment	Values
N	5, 10, 15, 20, 25, 30
Density	0.05, 0.1, 0.15, 0.2, 0.3, 0.5, 0.8
Reciprocity	0.0, 0.25, 0.5, 0.75, 1.0
Tweak Rate	0.005, 0.01, 0.05, 0.1, 0.15, 0.25
Coloring Condition	0: 1-Colored digraphs
	1: 2-Colored digraphs
Matrices per condition:	25

^a Some densities precluded some levels of reciprocity; in these cases, the closest possible value was used.

Table 1: Conditions for the Virtual Experi	ment
--	------

	you the state isoperiment
Treatment	Values
N	5, 10, 15, 20, 25, 30
Density	0.05, 0.1, 0.15, 0.2, 0.3, 0.5, 0.8
Reciprocity	0.0, 0.25, 0.5, 0.75, 1.0
Tweak Rate	0.005, 0.01, 0.05, 0.1, 0.15, 0.25
Coloring Condition	0: 1-Colored digraphs
	1: 2-Colored digraphs
Matrices per condition:	25

^a Some densities precluded some levels of reciprocity; in these cases, the closest possible value was used.

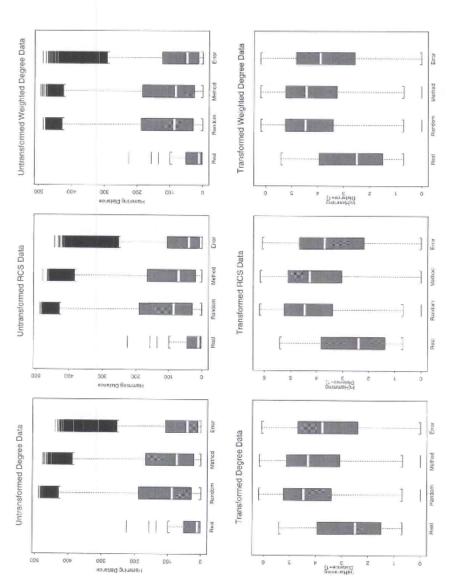


Figure 9. Boxplots of Hamming Experiment Data

<u>Y</u>	<u>X</u>	₿	R^2	Residual SE	n	ρ
Struct Dist ^a	Deg Method	0.3339***	63.66%	37.51	63000	0.6731
Struct Dist	RCS Method	0.3359***	64.01%	37.36	62692	0.6776
Struct Dist	WDeg Method	0.2909***	57.81%	40.42	63000	0.6061
Struct Dist	Rand Method	0.2854***	57.44%	40.59	63000	0.6010

Table 3: Prediction	f Observed Hamming	Distance by Condition
---------------------	--------------------	-----------------------

Coefficients	Log(Deg+1)	Log(RCS+1)	Log(WDeg+1)	Log(Rand+1)
Intercept	-2.4156	-2.3686	-2.5059	-1.9926
Log(N+1)	2.0213	1.9683	2.1643	2.0376
Log(Den+1)	1.1617	1.1380	1.4358	1.3608
Log(Recip+1)	-0.1802	-0.2440	-0.2123	-0.1685
Log(StrDist+1)	0.2234	0.2619	0.1360	0.1023
log(Color+1)	-0.1158	-0.1125	-0.1408	-0.0508
R ²	91.11%	89.74%	91.83%	93.44%
Residual SE	0.4230	0.4624	0.4028	0.3304
	All regressions and co	pefficients significant be	clow the 0.0001 level.	0.0001

Table 4: Diagnostic Error Regression

100 000 1		te 4. Diagnosiie Elitoi	Regression	
Coefficients	Log(Deg Error+1)	$Log(RCS\ Error+1)$	Log(WDeg Error+1)	Log(Rand Error+1)
Intercept	-3.8347	-3.8577	-3.7440	-2.9404
Log(N+1)	2.4954	2.5054	2.4972	2.2446
Log(Den+1)	1.9869	1.9811	2.3205	2.2113
Log(Recip+1)	-0.3236	-0.4064	-0.3446	-0.2766
Log(Color+1)	-0.1460	-0.1435	-0.1957	-0.0774
\mathbb{R}^2	79.67%	78.50%	80.95%	80.06%
Residual SE	0.7197	0.7483	0.6987	0.6487
	All coc	fficients significant below	the 0.0001 level.	

Table 5: T-tests of Adjusted Errors Across Methods

Method Deg	Mean Adj Error 0.1619	Method-Deg	Method-RCS t=6.2248 ^{b***}	Method-WDeg t=-45.5296c+**	<u>Method-Rand</u> ¹¹ t=-153.3434 ^{c***}
RCS	0.1582	t=-6.2248b***		t=-51.1143 ^{b***}	t=-
WDeg	0.1909		t=51.1143 ^{b***}		152.4636 ^{b***} t=-76.9855 ^{c***}
,	All Rand comparisons	reflect paired obse	ervations. b df=1250	590 ° df=125998 **	* p<0.001

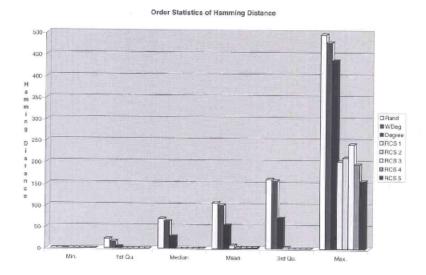


Figure 10. Order Statistics of Observed Hamming Distance for Isomorphic Graph Experiment

Table 6: Observed Hamming Distances for Is	somorphism Experiment
--	-----------------------

		THE RESIDENCE	S Distances je	1 Isomorphis	m Experiment	
Method	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Rand	0	22	70	107.5	162	496
WDeg	0	16	64	101.7	158	478
Degree	0	4	28	53.97	72	437
RCS 1	0	0	0	8.183	4	204
RCS 2	0	0	0	3.078	0	212
RCS 3	0	0	0	2.787	0	242
RCS 4	0	0	0	2.694	0	196
RCS 5	0	0	0	2.841	0	158

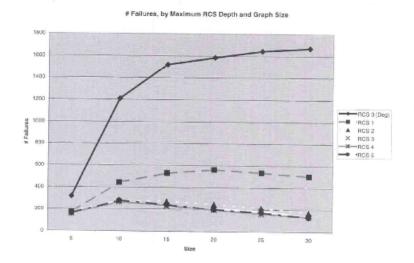


Figure 11: Total Number of Labeling Failures for Isomorphic Graph Experiment

Table 7: Number of	f Failures under	RCS (ar	d Degree)
--------------------	------------------	---------	-----------

	RCS Search Depth						
N	<u>o</u>	1	2	3	4	5	
5	318	172	177	155	156	158	
10	1209	444	282	270	268	280	
15	1521	529	275	231	236	244	
20	1586	562	249	205	196	199	
25	1645	536	217	187	161	172	
30	1671	504	170	135	135	130	

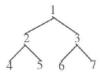


Figure 12. Central Graph for Four Sample Graphs

Table 8: Conditions for the Central Graph Experiment

Constraint N Density^a Values 5, 9, 13 Density^a
Coloring Condition 1-Coloreu Hause
Matrices per condition: 100

a The N=9, d=0.75 condition was omitted due to computational constraints.

Table 8: Central Graph Analysis - Cumulative Results

2.000	or community or t						
Ham Dist	Rand	Deg	RCS	WDeg			
Mean	27.62875	24.57875	24.56625	26.19875			
Stdev	21.91383567	18.9185408	18.963125	20.6012685			
Skew	0.811268223	0.54034	0.5401834	0.69140274			
T-Test*	Rand	Deg	RCS	WDeg			
Rand							
Deg	p<0.0001						
RCS	p<0.0001	p<0.86					
WDeg	p<0.0001	p<0.0001	p<0.0001				
N=800, m=100, n=5,9,13, d=0.25, 0.5, 0.75, Paired observations							

This document was cre The unregistered version	eated with Win2PDF avo	ailable at http://www.da /aluation or non-comme	neprairie.com. ercial use only.