CURRENT RESEARCH IN SOCIAL PSYCHOLOGY

AN ALGORITHM TO GENERATE CONNECTED GRAPHS*

John Skvoretz
University of South Carolina

ABSTRACT

   Connected graphs hold special interest for network exchange
researchers because all experimentally-studied networks are connected
graphs. Competing theoretical proposals differ in the predictions they
make for some sets of connected graphs, but not for other sets. Yet
researchers do not have at their disposal any catalogue of connected
graphs for even relatively small-sized groups. In this article, I
review the theoretical basis for an algorithm that generates all
connected graphs of a particular size and provide a pseudo-code version
of the algorithm itself. I then address some considerations regarding
its use in the network exchange field.

                          [page 43]
- -
                          [page 44]

INTRODUCTION

   All of the exchange networks studied experimentally in the field of
network exchange research are connected graphs, wherein the nodes are
positions and the lines are exchange relations (i.e., relations in
which agreement on the terms of exchange produces mutual benefit).
Further, all positions are connected to one another either directly or
indirectly. This connectedness of the set of positions is essential
since the very nature of network exchange research focuses on how
indirect connections affect the relative power of actors who are
directly connected to one another. The graphs vary in size, with six
positions being the largest for which published data exist (Skvoretz
and Willer 1993).

   Network exchange research presently harbors several contending
theories of power distribution. Evaluating these theories involves many
considerations, including parsimony in the assumptive base, breadth of
coverage, predictive precision, and embeddedness in accepted

explanatory frameworks (e.g., rational choice theory). One important
consideration is empirical accuracy: does one proposal make more
accurate predictions than the others? For many graphs, the contenders
make very similar predictions and so they cannot be used to assess
accuracy. Rather, research must use those graphs (if any) for which the
alternatives make different predictions. Currently, discovery of such
candidates is a hit-or-miss proposition as there exists no accessible
catalogue of connected graphs with which to systematically search for
"critical" networks.

In addition to the evaluation of empirical accuracy, a catalogue of
connected graphs could aid in the development of particular approaches,
as can be exemplified by Lovaglia, Skvoretz, Markovsky, and Willer
(1995). These authors use a series of "thought experiments" to refine
the Graph Power Index (GPI) approach first proposed by Markovsky,
Willer, and Patton (1988): Refinements are proposed to solve
problematic networks and then evaluated by inspecting the predictions
offered for other networks of similar size or configuration. Only those
refinements that offer sensible predictions are retained for
empirically-based evaluation. Hence, a catalogue of connected graphs is
needed grist for the "thought experiment" mill.

- -

ALGORITHMS FOR GENERATING GRAPHS

The problem of generating all connected graphs for a set p of points
is a combinatorial problem. One procedure for generating combinatorial
configurations of a certain type is called the "classical method" by
Read (1978). In this procedure, there are two lists: one of old
configurations and one of new configurations. Potential candidates for
the new list are created by performing an operation on an old
configuration, e.g., adding a node and an edge to one of the points in
the old configuration. The candidate is added to the new list only if
it is not already present. Doing this, however, requires checking the
candidate against each of the items already present on the new list, a
time-consuming process insofar as there are usually many isomorphic
versions of any item on the new list, and each of these versions must
be detected and discarded. For instance, the three networks in Figure 1
are isomorphic (i.e., they exhibit the same structure). Therefore, on
any list of connected graphs of size 4, only one of them can appear. As
the size of the graph and the number of edges grow, the combinatorial
possibilities explode.

```
     A - B          B - C          A - C
      \ /            \ /            \ /
       C              A              B
       |              |              |
       D              D              D
```
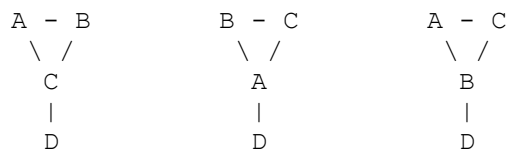
Figure 1. Three isomorphic graphs.

Read (1978) proposes a general algorithm that does not require
looking back over the new list to detect and discard duplicates. In
this procedure, as each candidate is produced it can be immediately

tested for inclusion on the new list by simply inspecting the configuration itself. Hence, Read titles his paper "Every One a Winner" because the candidate is added to the new list only if it is a winner. A principle advantage of this algorithm is that the list of new elements can be stored externally since searches of it are never required.

Read's "orderly" algorithm may be described as follows. We must first specify which of all possible isomorphic configurations will represent its class in the list of configurations. This particular configuration is called the "canonical configuration." The center graph in Figure 1 is the canonical configuration for its isomorphism class. Canonicity is defined by representing the upper right triangle of the network's adjacency matrix as a binary integer, concatenating the first row of this triangle to the second, both to the third, and so on. The canonical configuration for the class is the one that produces the largest binary integer or code. The codes for the networks in Figure 1 are 110101, 111100, and 110110, the second one being the largest.

- -

Second, we must specify an ordering of the list of canonical configurations. The binary integers corresponding to each canonical configuration form a logical basis for ordering the list. In fact, we arrange the list in descending order by these integers.

Finally, we must define an augmenting operation by which one element from an old list can produce an ordered sequence of potential elements on a new list in the order they would appear on the new list if they were found to be acceptable candidates. The operation we use takes the binary code of a canonical configuration of p nodes and q edges and produces potential candidates for the list of p nodes and q+1 edges as follows. The binary code of the input configuration is scanned from right to left to find the rightmost non-zero element. If the code is 111100, this is the fourth element. Then the candidate configurations are created by changing in turn each of the trailing 0's to 1, proceeding from left to right. Thus, from 111100 we obtain two candidates, 111110 and 111101.

The algorithm involves processing an old list of canonical configurations, taking each element in turn, and creating potential candidates for the new list. For each candidate, the algorithm must discover if it is a canonical configuration by inspecting permutations of rows and columns in the underlying adjacency matrix for an equivalent configuration with a larger code value. If this value ever equals or exceeds the code value of the last configuration added to the new list, the candidate is not a canonical configuration and processing can move on to the next candidate. If the configuration is canonical and its code value is less than the new list minimum, it is a "winner," allowing it to be added directly to the new list and the new list minimum to be updated.

The program that actually generates all connected graphs of a given size is presented in pseudo-code form in Table 1. Note that it generates all graphs whether connected or not and then retains only

those that are connected. The algorithm is not effective if its
operation is confined to just connected graphs. There are connected
graphs with q+1 edges whose code number is greater than other connected
graphs with q+1 edges but whose production from the ordered list of
canonical connected graphs with q edges will occur after the production
of those graphs with lower code numbers. Consequently, they will never
be added to the new list. However, they will be produced before these
other graphs if the augmentation operation applies over all graphs,
connected or not.

- -

```
input nodes
initialize file "g00"
        with bit-string of 0's {length =[nodes(nodes-1)]/2}
for edges = 0 to [nodes(nodes-1)]/2 - 1
  open file "g{nodes}{edges}" for input as #1
  open file "g{nodes}{edges+1}" for output as #2
  open file "g{nodes}{edges+1}c" for output as #3
  initialize current minimum code value
  while not end-of-file #1
    read string from #1
    find rightmost non-zero element of string
    for each trailing zero
      change to 1 and create new adjacency matrix
      for each permutation of rows and columns
        compare binary number of permuted adjacency matrix with current
                minimum and if greater, try next trailing zero
        compare binary number of permuted adjacency matrix with current
                maximum and if greater, save as new maximum
    set current minimum =3D maximum
    write maximum to #2
    if connected, write maximum to #3
  wend
close #1, #2, #3
end
```

Table 1: Pseudo-code for an orderly algorithm
          to generate connected graphs

     Even though the orderly algorithm is relatively more efficient than
the "classical" approach, it still becomes time consuming as the number
of nodes increases. On a 90mhz Pentium it will take about 2 hours to
generate all the connected graphs over 7 nodes, where one finds 853
non-isomorphic connected graphs. Over 8 nodes, there are 11,118
connected graphs. Consequently, generating the catalogue of these
graphs takes literally days of CPU time. Thus, it is unreasonable for
each researcher to generate his or her own catalogue of connected
graphs. Instead, the catalogue, once generated, should be made widely
accessible. Consequently, I supply an Appendix which catalogues all
connected graphs from size 2 to size 7 and can supply upon request the
catalogue of size 8 connected graphs. A Sun SparcStation is currently
at work (on a part-time basis since December) generating all size 9
connected graphs.

- -

CONCLUSION

   In addition to advancing theoretical inquiry, the catalogue of
connected graphs can standardize nomenclature for experimental
networks. For instance, the graph in Figure 2 is called "KITE" in the
work of Markovsky, Skvoretz, Willer and colleagues (Markovsky,
Skvoretz, Willer, Lovaglia, and Erger 1993; Skvoretz and Willer 1993)
and "HOURGLASS" in the work of Bienenstock and Bonacich (1993). The
catalogue provides a canonical name for this graph: namely, the number
of nodes, followed by the number of edges, followed by its location in
the canonical ordering. Thus the graph in Figure 2 has the canonical
name "p5e6.2" because it is the second graph in the canonical listing
of all connected graphs with 5 nodes or points and 6 edges.

```
            B - C
             \ /
              A
             / \
            D - E
```
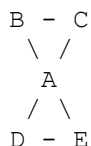
      Figure 2: The connected graph p5e6.2

   The catalogue also provides a canonical way of labeling the nodes of
an entry. One simply assigns letters to nodes in alphabetical order and
then builds the adjacency matrix as specified by the bit string, thus
standardizing how graphs are drawn for research reports.

   Both uses of the catalogue become more salient as the size of the
networks investigated becomes larger. Investigators from different
approaches can more easily communicate with one another over
problematic graphs since, in switching to a standardized naming system,
we avoid "pet" names that reflect accidental circumstances surrounding
an approach's discovery.

  Thus, the actual use of the catalogue appears to depend on (a)
maintaining theoretical heterogeneity in exchange network research, (b)
extending empirical and theoretical investigation to relatively large
networks (say, 7, 8 and 9 node networks) and (c) making accessible an
organized catalogue of connected graphs.

  The track record of the network exchange research field over the last
10 years suggests condition (a) will hold for the foreseeable future.
The present paper guarantees the satisfaction of condition (c). And the
internal dynamic of theoretical development via conjecture and
refutation pushes the explanatory envelope in the direction consistent
with condition (b). Only empirical extension may lag for the usual
budgetary reasons.

- -

ENDNOTE

*Reviewers provided helpful comments and suggestions. Address correspondence to: skvoretz-john@sc.edu

REFERENCES

Bienenstock, E.J. and P. Bonacich. 1993. "Game-Theory Models for Exchange Networks: Experimental Results." Sociological Perspectives 36: 117-35.

Lovaglia, M., J. Skvoretz, B. Markovsky, and D. Willer. 1995. "Assessing Fundamental Power Differences in Exchange Networks: Iterative GPI." Current Research in Social Psychology 1(2): 8-15, http://www.uiowa.edu/~grpproc.

Markovsky, B., J. Skvoretz, D. Willer, M. Lovaglia and J. Erger. 1993. "The Seeds of Weak Power: An Extension of Network Exchange Theory." American Sociological Review 58: 197-209.

Markovsky, B., D. Willer, and T. Patton. 1988. "Power Relations in Exchange Networks." American Sociological Review 53: 220-36.

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

Skvoretz, J. and D. Willer. 1993. "Exclusion and Power: A Test of Four Theories of Power in Exchange Networks." American Sociological Review 58: 801-18.

AUTHOR BIOGRAPHY

John Skvoretz is a Carolina Distinguished Professor of Sociology at the University of South Carolina. Current research projects concern formal models of social networks, status and participation in discussion groups, power in exchange networks, and theoretical studies of action structures. He is the editor of Connections, the bulletin of the International Network for Social Network Analysis and an Associate Editor of the Journal of Mathematical Sociology.