

CURRENT RESEARCH IN SOCIAL PSYCHOLOGY

Volume 8, Number 16
Submitted: August 7, 2002
First Revision: October 2, 2002
Second Revision: February 13, 2003
Third Revision: March 3, 2003
Accepted: March 5, 2003
Publication Date: March 5, 2003

OPTIMAL SEEK SIMPLIFIED

Dudley Girard
University of South Carolina

Casey A. Borch
University of Connecticut

ABSTRACT

Finding network breaks is one area of research in network exchange theory. A network break occurs when one or both positions sharing a relation do not to use that relation. Optimal Seek locates network breaks by assigning each position a scalar probability of being included in an exchange. This method is shown to be computationally expensive and therefore restricted to small networks. In this paper, we develop a computationally inexpensive method of determining a position's probability of being included in an exchange. We utilize this method to optimize Optimal Seek. This new optimized form is termed Optimal Seek Simplified.

[225]

[226]

INTRODUCTION

Research in network exchange theory has demonstrated that power exercised by actors is conditioned by structure.[1] In this tradition, a person or collectivity, termed actor, is connected, via a relation, to one or more other actors, forming a network. In experiments to examine exchange networks, actors typically enter into negotiations over resource pools placed between actors. Exchange occurs when actors have agreed how to divide the resource pool. In economic terms, actors in these kinds of exchange networks play a zero-sum game, meaning that when one actor gains resources at least one other loses an equal amount of resources. The actor with the larger part of the resource pool is said to exercise power over the other. That is, the greater an actor's share of the resource pool, the higher that actor's positional power is predicted to be (Dahl 1957; Lukes 1974; Zelditch 1992). We necessarily limit the discussion to networks in which each position can maximally make one exchange and all resource pools are equal.

Beginning with Markovsky et al. (1988), network breaks have become an important area of research in network exchange theory. A network break occurs when one or both positions sharing a relation do not to use that relation. An early method of locating breaks in networks, utilizing the Graph-theoretic Power Index (GPI) (Markovsky et al. 1988), is restricted to certain network types. Following GPI, the Core (Bienenstock and Bonacich 1993) was introduced. It locates network breaks, but is severely scope limited. The most recent method, Optimal Seek (Simpson and Willer 1999), is inadequate due to high computational cost. We review each of these methods in the next section.

A new method, Optimal Seek Simplified, greatly reduces the computational cost of Optimal Seek by treating excludability as a binary rather than a scalar variable; either an actor is excludable or he/she is not. An actor is excludable if there exists some *pattern of exchange* that leaves that actor without an exchange partner; otherwise he/she is non-excludable. Let pattern of exchanges mean the total number of possible actor pairings within a given network. For example, the four-actor line network (A - B - C - D) has two patterns of exchange: [A - B] [C - D] and [B - C]. In the latter, A and D are excluded. Thus, actors A and D are excludable, while actors B and C are non-excludable.

According to Optimal Seek Simplified, recognizing an actor to be either excludable or not excludable is all that is needed to locate breaks in networks. As will be seen, this assertion does not lead to a decrease in the predictive power of the method. Additionally, Optimal Seek Simplified clarifies the locating of *power components* within networks. Although Simpson and Willer (1999) do not define "power components," we infer from their paper that power components are smaller networks comprised of nodes and relations that are part of the overall network (i.e. subnetworks). This last point is fully explained in the remaining sections.

This paper is separated into four further sections. The first section reviews previous research and the evolution of network breaks. In the second, we present the algorithm used by Optimal Seek Simplified to locate network breaks. The third section compares Optimal Seek and Optimal Seek Simplified focusing on the computational cost of each, while the fourth offers a conclusion.

THE EVOLUTION OF NETWORK BREAKS

The Graph-Theoretic Power Index

Theories that find breaks in exchange networks are not new. Markovsky et al. (1988) use Graph-theoretic Power Index (GPI) values to determine the relative power of network positions. They theorize that some actors favor certain relations over others depending on their power relative to that of their neighbors. Thus, GPI can be used to predict the relations an actor is more likely to use, and, as a result, network breaks. Briefly, by counting path lengths, GPI calculates the relative power of all positions in a given network. Paths of even length are considered to be disadvantageous, while paths of odd length are considered to be advantageous, and only non-intersecting paths are counted. For example, consider the A - B - C network. A has 1 one path [A - B] and 1 two path [A - C]. Thus, A's GPI value is $1 - 1 = 0$, similarly for C. Since B has 2 one paths [B - A] and [B - C], its GPI value is 2. Therefore, according to GPI, B is predicted to be the most powerful position in the A - B - C network.

[226]

[227]

Markovsky et al. (1988:225) develop four axioms that are used to determine which actors seek exchange with which, and, as a result, where networks break.

Axiom 1: Compute relative positional power using GPI.

Axiom 2: i seeks exchange with j if and only if i 's power is greater than j 's, or if i 's power relative to j equals or exceeds that of any of i 's other relations.

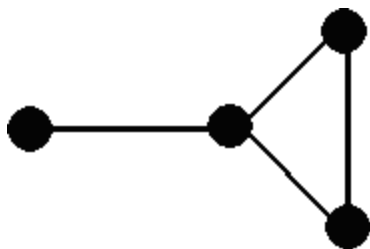
Axiom 3: i and j can exchange only if each seeks exchange with the other.

Axiom 4: If i and j exchange, then i receives more resources than j if and only if i has more power than j .

After much additional work, two problems were found with GPI. First, it was found that it failed to correctly locate power positions in certain network types (for a review see Willer 1999:113-116). To correct this problem, Markovsky et al. (1993) develop a method of calculating positional power based on a position's likelihood of being included in an exchange. This method is termed Exchange Seek Likelihood (ESL). Briefly, ESL calculates the likelihood of being included by assuming that each actor seeks to exchange with all connected others with equal probability.

Second, certain networks such as the Stem, see Figure 1, oscillates between having breaks and not having breaks (for a review see Willer 1999:113-116). Unfortunately, ESL fails to solve the oscillation problem, as the problem lays in the underlying method of how breaks are determined.

Figure 1. The Stem Network.



[227]

[228]

The Core

Bienenstock and Bonacich's (1993) Core is based on the idea that: "No group of players will accept an outcome if by forming a coalition they can do better" (Beinenstock and Bonacich 1993:125). By applying this idea to exchange networks, researchers are able to predict which exchanges will occur in one-shot games. In this context, a one-shot game refers to networks in

which actors exchange maximally once during the entire game. For an example of the Core, consider the A - B - C - D network, in which pools of 24 resources are placed between each actor. If B and C exchange then only 24 of the total 72 resources are divided. However, if A and B exchange and C and D exchange then 48 of the total 72 resources are divided. Thus, the Core predicts that A, B, C, and D favor the latter exchange outcome over the former as it utilizes more of the total resources. As such, the Core predicts in a one-shot game that, when each node can make only one exchange, B and C never exchange. This is a break by the definition given above: a relation where no exchange is predicted to occur is a break.

Because of the method by which the Core determines which exchange outcomes are possible, all "suboptimal" relations become breaks (Bienenstock and Bonacich 1993). A relation is suboptimal if and only if exchanging in that relation necessarily reduces the maximum number of exchanges possible for the network (Simpson and Willer 1999:273). The problem with the Core is that in repeated games - the condition under which most network exchange research is conducted - not all suboptimal relations are breaks (for a review see Willer 1999: 275,282-283). However, it would be another five years before a solution to this problem was developed.

Optimal Seek

To find network breaks in repeated games, Simpson and Willer (1999) build on the work of Bienenstock and Bonacich (1993) and Markovsky et al. (1993). They anchor their method in the following four definitions of network power (Simpson and Willer 1999:271-272):

D1: A network is strong power if and only if the number of non-excludable actors is less than the number of excludable actors, excludable positions connect only to non-excludable positions, and non-excludable positions connect only to excludable positions.

D2: A position is considered strong high power if it is non-excludable in a strong power network.

D3: A network is equal power if and only if all positions are excluded at the same rate.

D4: Weak power networks are defined as networks that are neither strong power nor equal power and are divided into two types:

a) A Type 1 weak power network is defined as a weak power network in which at least one position is never excluded.

b) A Type 2 weak power network is defined as a weak power network in which all positions are excludable, but not at the same rate.

[228]

[229]

Simpson and Willer use their new method, called Optimal Seek (OS), to derive the following six hypotheses (1999:279-282):

H1: If i and j are strong high power positions, they will not exchange: the network breaks at the i - j relation.

H2: If j is a strong high power and i is the highest power position in a Type 1 weak power network, they will not exchange: the network breaks at the i - j relation.

H3: If j is strong high power and i is not excluded in an equal power network, they will not exchange: the network breaks at the i - j relation.

H4: If j is strong high power and i is ever excluded in a weak power network, they will exchange: the network does not break at the i - j relation.

H5: If j is strong high power and i is excluded in an equal power network, they will exchange: the network does not break at the i - j relation.

H6: Weak power networks do not break.

OS locates breaks in networks by applying the following algorithm (Simpson and Willer 1999:276):

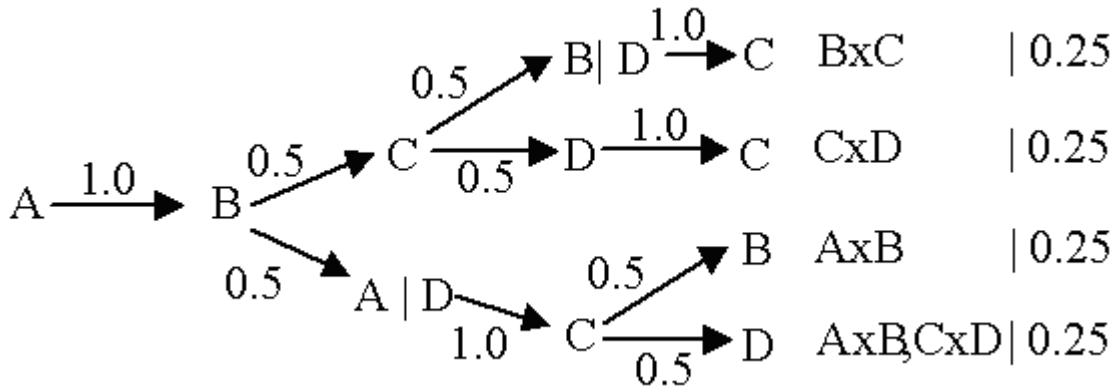
- 1) Apply ESL to find (a) strong, weak, and equal power components, and (b) suboptimal relations.
- 2) Remove all suboptimal relations that connect high power positions to positions outside the strong power component. Recalculate ESL for the resultant subnetworks.
- 3) A suboptimal relation is a break iff the outside position is never excluded after the relation is removed. A suboptimal relation is not a break if the outside position is excludable after the relation is removed.

We do not dispute the network breaks identified by OS. We do, however, take issue with OS's method of finding suboptimal relations and power components (Step 1). The method used, ESL, is problematic when applied to large networks. An understanding of why ESL is a problem is found in the time complexity literature of computer science. Time complexity deals with determining how many steps a computer will need to complete a specific task. For example, the task of counting the number of nodes in a network takes one time step for each node in the network, thus its time complexity would be $O(n)$ where n is the number of nodes in the network. If an algorithm's time complexity is $O(n)$, then it will take at most n steps to complete (Smith 1989).[2]

OS and the Problems of ESL

To understand the time complexity of computing ESL values, it is important to understand how the values are computed. The easiest way to explain this is with a tree graph that shows how actors are predicted to seek exchange with each other (Figure 2).

Figure 2. The ESL Probability Tree for the 4-Line (A - B - C - D).



(Note: Arrows indicate an actor looking in a specific direction. The vertical lines indicate that an actor can look no further on its specific path. The number on the arrow is the probability that an actor is looking that way. At the end of each chain is the list of exchanges that may occur and the probability of each occurrence.)

As can be seen, all possible exchanges are not accounted for, however, it should be obvious that after A - B exchange C - D must exchange and, of course, the reverse is also true. In short, the time complexity for constructing the ESL probability tree, without representing each branch, is:

$$O(D(1) \cdot D(2) \cdot \dots \cdot D(n)) \quad (1)$$

Where n is the number of actors in the network and $D(i)$ is the actor i 's "degree." An actor's degree is the number of connections (relations) that actor has to other actors in the network. To be complete, if any branch of the tree does not cover all possible exchanges, then subsequent ESL trees must be built to account for all remaining exchanges.

For an example of ESL's computational cost, consider a network of 30 nodes with each node having 3 connections (degree = 3). In this case, the computational cost to build the *initial* ESL tree is: 2.06×10^{14} . Any additional trees will add to this cost. In fact, for this example, there are approximately 6.29×10^{23} exchange paths in the complete ESL tree. Additionally, this number does not include the costs of running ESL a second time in Step 2 of OS. Simply put, ESL is applicable only to small networks. This issue is revisited after reviewing Optimal Seek Simplified.

In the next two sections, we present the algorithm used by Optimal Seek Simplified (OSS) to locate network breaks, followed by a comparative look at the time costs for OS and OSS.

OPTIMAL SEEK SIMPLIFIED

The Optimal Seek Simplified (OSS) method introduced here solves the two main problems with Optimal Seek (OS). First, OSS provides a clear method to locate power components. Second, OSS replaces ESL with a cost efficient way to find power components, suboptimal relations, and network breaks. To do this, OSS conceptualizes excludability as a binary rather than as a scalar variable. Thus, unlike OS, Optimal Seek Simplified does not calculate a position's specific probability of inclusion by using a computationally expensive method such as ESL; rather it labels positions as excludable (E) or non-excludable (I).

To determine whether a node (i) is an I node or an E node, one need only find if there is some pattern of exchange by which i can be excluded by its neighbors. A neighbor is considered to be any node connected to the node in question (see Figure 3).

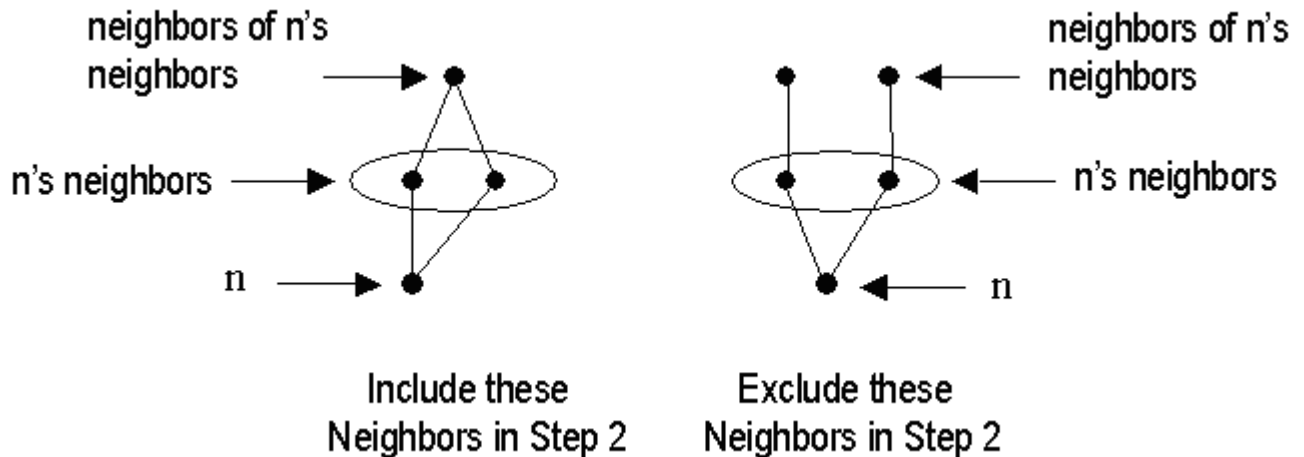
The algorithm for determining excludability is as follows (this method replaces ESL):

- 1) For any node n ,
 - a) If no neighbors of n over-lap with one or more neighbors of n 's neighbors (excluding n , of course), then n is excludable. Label n "E" and stop.
 - b) If any of n 's neighbors have only n as a neighbor, then n is not excludable. Label n "I" and stop.
- 2) If a pattern of exchange is found by using n 's neighbors as starting points in which n cannot exchange, then n is excludable. Label n "E" and stop.
- 3) If no pattern of exchange is found to satisfy step 2, then n is not excludable. Label n "I" and stop.

[231]

[232]

Figure 3. Illustration of Step 1a.



Because OSS uses a binary rather than a scalar representation of exclusion, we must reformulate the definitions of equal power and Type 2 weak power networks provided by OS. Due to ESL's method of computing scalar probabilities, the only way a network of E nodes will each have the same scalar value of inclusion is if they all are looking at each relation at the same rate. Thus, for a network of E nodes to have the same scalar value of inclusion they must all have the same degree. Therefore, a network is equal power if all nodes are I nodes or if all nodes are E nodes *and* each has the same degree. In addition, a Type 2 weak power network is redefined as a network of all E nodes in which at least one node has a different degree.

[232]

[233]

Finding Power Components

Before getting to OSS's algorithm, we must first address the problem of finding power components. Optimal Seek does not give a specific method other than: "Apply ESL to find (a) strong, weak, and equal power components" (Simpson and Willer 1999:276). OSS, on the other hand, replaces this with Steps 1 - 5 of OSS:

- 1) First, using the algorithm introduced above, label all nodes that cannot be excluded as I and those that can be excluded as E.
- 2) Any relation that connects an E node to an I node, where the E node also has a connection to another E node, is removed. These relations are termed *class 1* relations.
- 3) Remove relations between I nodes if at least one of the I nodes is connected to two or more E nodes. These relations are termed *class 2* relations.
- 4) Label the nodes of the network using the algorithm above and used in Step 1.

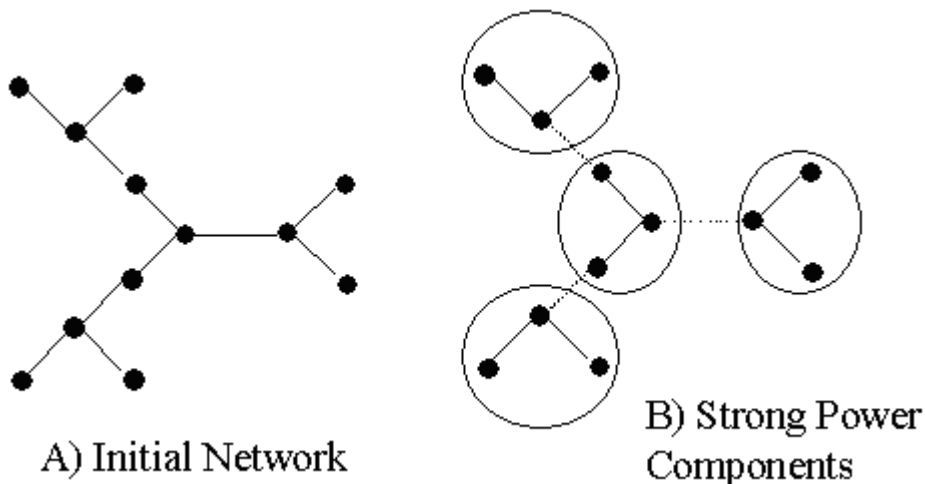
5) Using the original definitions of strong and Type 1 weak power networks and the redefined terms for equal and Type 2 weak power networks, identify the power components that are strong power networks.[3]

Before continuing we have some additional comments concerning the locating of power components in networks. As just seen, it is possible to identify strong power components. However, as shown in Figure 4 below OSS's method of identifying strong power components points to an oversight in OS (Simpson and Willer 1999). Simpson and Willer's hypotheses provide no guide to determine what happens to low power nodes of a strong power component that can reconnect to other nodes in the network. Additionally, their hypotheses provide no guide to determine what happens to high power nodes of strong power components if they become excludable. However, their algorithm does address what to do in such a situation (Simpson and Willer 1999: 276).

[233]

[234]

Figure 4. An Example Network with Four Strong Power Components.



Furthermore, Simpson and Willer (1999) leave questions concerning the differentiation of weak and equal power components in networks unanswered. Both Type 1 weak power components and equal power components can have relations that connect two non-excludable nodes. Additionally, both Type 2 weak power components and equal power components can have relations that connect two excludable nodes. However, equal power components cannot have relations connecting excludable nodes to non-excludable nodes, while Type 1 weak power components can. Therefore, if all *class 1* relations - excludable nodes connected to non-excludable nodes in which the excludable nodes also have at least one connection to an additional excludable node - were removed (after first removing the strong power components, of course), then the network would seem to be properly divided into its various power components. However, this may not be the case. Applying the preceding logic to the Stem network (Figure 1), for example, we find that it breaks into two dyads. Is this then a correct designation of the power components for the Stem or should the component consist of the

complete network? Simpson and Willer (1999) leave this problem unresolved. However, since this problem has no direct effect network breaks we leave it to be addressed in future work.

[234]

[235]

Finding Suboptimal Relations

Because in one-shot games all suboptimal relations are breaks (Beinenstock and Bonacich 1993), we need a means to locate all such relations. Thus, we propose the following four statements:

OSS1: Any relation that connects I nodes where each of the I nodes is also connected to one or more E nodes is a suboptimal relation.

OSS2: Any class 1 relation is a potential suboptimal relation.

OSS3: Any relation that connects E nodes is not a suboptimal relation.

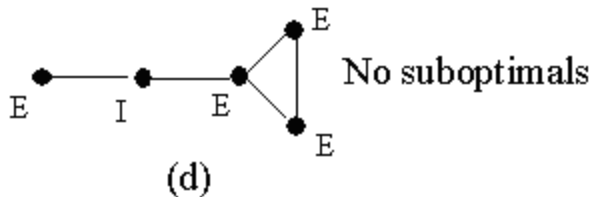
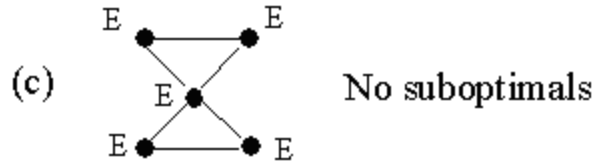
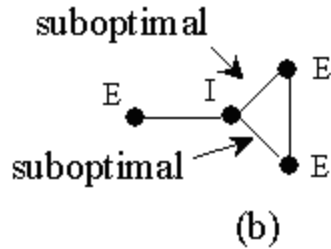
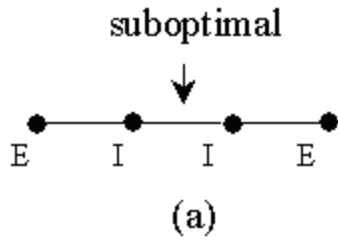
OSS4: After all suboptimal and potential suboptimal relations are removed; any relations removed from nodes that are now E nodes are not suboptimal relations (this point is fully discussed below).

There is not space enough to present a formal proof of the statements, however, a set of results using the statements is provided in Figure 5.

[235]

[236]

Figure 5. Illustration of OSS1, OSS2, OSS3, and OSS4.



(Note: Network (a) shows the use of OSS1. Network (b) shows the use of OSS2. Network (c) shows the use of OSS3. Lastly, Network (d) shows the use of OSS4.)

[236]

[237]

Utilizing OSS1 - OSS4 we replace Step 1b of OS with Steps 6 - 9 of OSS:

- 6) Using the network and labels from Step 1, remove all *class 1* relations.
- 7) Remove relations between I nodes if each I node is connected to an E node. These relations are termed *class 3* relations.
- 8) Label the new graph using the method from Step 1.
- 9) If an E node had a relation removed and it is still an E node, then that relation is placed back. Any remaining removed relations are suboptimal relations.

Replacing Step 2 of OS is Step 10 of OSS:

- 10) Using the labels from Step 5, remove all suboptimal relations that connect two nodes where at least one of the nodes is an I node in a strong power component and the other node is an I node from the initial network.

Replacing Step 3 of OS are Steps 11 - 13 of OSS:

- 11) Label the nodes using the method from Step 1.
- 12) Add back any removed suboptimal relations where one of the nodes is now an E node.
- 13) Repeat Steps 11 and 12 until no suboptimal relations are added back.

The algorithm is now complete. In the next section, we offer a comparison of the computational costs associated with OS and OSS.

[237]

[238]

A COMPARISON

Before comparing OS to OSS, we first need to discuss the computational cost inherent to OSS. The main computational cost involves discovering whether a position is excludable or not. This is done, for any given position, by determining if there is some pattern of exchange in which that position is excludable. The computational cost of the algorithm for determining if a node is an E node or an I node (see the previous section) is as follows:

The time complexity of Step 1 is:

$$O(m \cdot \log m) \quad (2)$$

Where m is the sum of the degrees of the neighbor nodes.

The time complexity of Step 2 is:

$$O(k^2) \quad (3)$$

Where k is the sum of the degrees of the remaining neighbor nodes.

OSS does not have to examine all patterns of exchange, because it needs only to uncover if there exists a pattern of exchange where at least one neighbor of a node cannot exclude it. For this reason, the OSS method for finding excludability is computationally efficient.

An Example

Consider a conservative example: A network contains 50 nodes in which each node has an average degree of 2.5. Thus, there are 50 nodes in the network and 125 relations. For OS, the cost of finding suboptimal relations and power components is approximately 4.6×10^{42} steps. To get some idea of what this number means, a Pentium IV takes about 1 second to execute an addition operation 1×10^8 times. Thus, to execute the same operation 4.6×10^{42} times would take about 4.6×10^{34} seconds, or roughly 1.4×10^{27} years. If computer speeds continue their

present rate of increase, then it will take nearly 243 years before a computer can execute such an operation in less than one year (Hennessy and Patterson 1990; SPEC benchmarks). For OSS, the cost to find the suboptimal relations and power components for this network is approximately 5000 steps or less than a second of computation time. Table 1 displays additional comparisons of time complexity for OS and OSS.

[238]

[239]

Table 1. Cost Comparison of OSS and OS.

Network Size in Nodes	OSS Excludability	OS Excludability (ESL)
50	5,000	4.6×10^{42}
75	7,500	7.4×10^{66}
100	10,000	2.0×10^{97}

(Note: Table 1 gives a cost comparison in steps. Networks have an assumed average degree of 2.5. The values given are approximates.)

CONCLUSION

To conclude, by using a parsimonious method of determining positional power, we use OSS, which is computationally inexpensive, to replace OS, which is computationally expensive. Additionally, assuming we fully represented Simpson and Willer's (1999) understanding of power components, we have removed ambiguity from OS by providing a concise algorithm for finding strong power components based on OSS's positional power ranking method (E or I). Lastly, we have shown how Optimal Seek could be modified to use the new positional power ranking method to find network breaks. This optimized form we termed Optimal Seek Simplified. It is our hope that by making the locating of network breaks easier to compute we extend this area of research to larger, more complex networks.

ENDNOTES

1. See Stolte and Emerson (1977); Cook et al. (1983); Markovsky et al. (1988); Molm (1990); Skvoretz and Willer (1993); Bienenstock and Bonacich (1993); Markovsky et al. (1993); Friedkin (1995); Lovaglia et al. (1995); Willer (1999); Lucas et al. (2001).

2. Within the time complexity literature there exists a designation for a lower bound of computational cost as well (Smith 1989). However, due to the nature in which ESL values are computed it is not possible at this time to provide such a value. Nor, for that matter to provide what would be considered an average value for computational cost. We assert, however, that ESL's average computational cost will be quite large and that our values are an accurate representation of this cost.

3. Because breaks in repeated games occur only between high power nodes of strong power components and other non-excludable nodes we need only locate the strong power components in a network.

[239]

[240]

REFERENCES

Baase, Sara. 1988. *Computer Algorithms: Introduction to Design and Analysis*. New York: Addison-Wesley Publishing Company.

Bienenstock, Elisa Jayne and Phillip Bonacich. 1993. "Game-Theory Models for Exchange Networks: Experimental Results." *Sociological Perspectives* 36:117-135.

Book, Ronald. 1994. "Relativizations of the P=? NP and Other Problems: Developments in Structural Complexity Theory." *Society for Industrial and Applied Mathematics Review* 36:157-175.

Cook, Karen S., Richard M. Emerson, Mary R. Gillmore, and Toshio Yamagishi. 1983. "The Distribution of Power in Exchange Networks: Theory and Experimental Results." *American Journal of Sociology* 89:275-305.

Cook, S. 1971. "The Complexity of Theorem-Proving Procedures." *Proc. 3rd ACM Symp. Theory of Computing* 151-158.

Dahl, Robert. 1957. "The Concept of Power." *Behavioral Science* 2:201-218.

Friedkin, Noah. 1995. "The incidence of Exchange Networks." *Social Psychology Quarterly* 58:213-222.

Hennessy, John L. and David A. Patterson. 1990. *Computer Architecture: A Quantitative Approach*. San Mateo, California: Morgan Kaufmann Publishers, Inc.

Lovaglia, Michael, John Skvoretz, David Willer, and Barry Markovsky. 1995. "Negotiated Outcomes in Social Exchange Networks." *Social Forces* 74:123-155.

Lucas, Jeffrey W., C. Wesley Younts, Michael Lovaglia, and Barry Markovsky. 2001. "Lines of Power in Exchange Networks." *Social Forces* 80:185-214.

Lukes, Steven. 1974. *Power: A Radical View*. London, UK: Macmillan.

Markovsky, Barry, David Willer, and Travis Patton. 1988. "Power Relations in Exchange Networks." *American Sociological Review* 53:220-236.

Markovsky, Barry, John Skvoretz, David Willer, Michael Lovaglia, and Jeffery Erger. 1993. "The Seeds of Weak Power: An Extension of Network Exchange Theory." *American Sociological Review* 58:197-209.

Molm, Linda D. 1990. "The Dynamics of Power in Social Exchange." *American Sociological Review* 55:427-447.

Simpson, Brent and David Willer. 1999. "A New Method for Finding Power Structures." Pp. 270-284 in D. Willer (ed.). *Network Exchange Theory*. Westport, CT: Praeger.

[240]

[241]

Skvoretz, John and David Willer. 1993. "Exclusion and Power: A Test of Four Theories of Power in Exchange Networks." *American Sociological Review* 58:801-818.

Smith, Jeffrey D. 1989 *Design and Analysis of Algorithms*. Boston: PWS-KENT.

SPEC benchmarks. 2002. <http://www.specbench.org/osg/cpu2000/results/cpu2000.html>. Standard Performance Evaluation Corporation.

Stolte, John and Richard Emerson. 1977. "Structural Inequality: Position and Power in Exchange Structures." Pp. 117-138 in R. Hamblin and J. Kunkel (eds.). *Behavioral Theory in Sociology*. New Brunswick, NJ: Transaction Books.

Waller, David (ed.). 1999. *Network Exchange Theory*. Westport, CT: Praeger.

Zelditch, Morris Jr. 1992. "Interpersonal Power." Pp. 994-1001 in E. F. Borgatta and

M. L. Borgatta (eds.). *Encyclopedia of Sociology*. New York: Macmillan.

AUTHORS' ACKNOWLEDGEMENTS

The authors thank David Willer, Brent Simpson, Eugene C. Johnsen, Ben Bryan and two anonymous reviewers for their many helpful comments regarding this paper. We also thank CRISP editor Lisa Troyer. Direct correspondence to Dudley Girard, girard@cse.sc.edu, Department of Sociology, University of South Carolina, Columbia, SC 29208.

AUTHOR BIOGRAPHIES

Dudley Girard, PhD, is a Research Associate at the University of South Carolina. His current research focuses on how actor perceptions can affect power within exchange networks and network self-organization. He is presently one of the main developers of Web-Lab (<http://weblab.moore.sc.edu>), a tool for experimental research of Sociology and Economics. His e-mail address is: girard@cse.sc.edu.

Casey A. Borch is a Ph.D. candidate at the University of Connecticut. He has recently co-authored a paper on group solidarity for the upcoming edition of *Advances in Group Processes*. His current research interests include actor perceptions of social networks, self-organization of social networks, and the application of complexity theory to social network research. His e-mail address is: Casey.Borch@huskymail.uconn.edu.

[241]

[242]