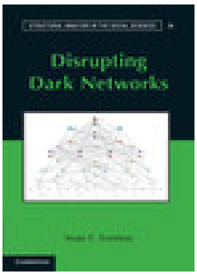


Disrupting Dark Networks

Sean F. Everton

Book DOI: http://dx.doi.org/10.1017/CBO9781139136877

Online ISBN: 9781139136877

Hardback ISBN: 9781107022591

Paperback ISBN: 9781107606685

Chapter

8 - Brokers, Bridges, and Structural Holes pp. 253-285

Chapter DOI: http://dx.doi.org/10.1017/CBO9781139136877.012

Cambridge University Press

8

Brokers, Bridges, and Structural Holes

8.1 Introduction

Betweenness centrality, which we examined in the previous chapter, implicitly introduced the concept of *brokerage*, which is the idea that some actors are more likely to control the flow of resources than others. In this chapter we explore the notion of brokerage in more depth. We begin by looking at Ron Burt's (1992a, b) notion of structural holes, which builds upon Mark Granovetter's (1973, 1974) work regarding weak ties. Burt argues that actors who sit on either side of bridges (i.e., ties) that span gaps in the social structure (i.e., structural holes) are in a position to broker the flow of resources through the network. Somewhat related to Burt's approach is bi-component analysis, which identifies the bridges and actors (i.e., cutpoints) that if removed, disconnect the network (Wasserman and Faust 1994:112–115). Although the notion that the dissolution of certain ties or the isolation of particular actors will disconnect a network is intuitively appealing, in well-connected networks, it is often difficult to find such actors and bridges. Their removal may isolate one or two actors, but it may not disconnect the network in a substantive way. However, we can identify *sets of actors*, that if removed, will either disconnect a network or substantially fragment it (Borgatti 2006). This is known as the *key player* approach, and we will examine it in Section 8.4. Similarly, by measuring the degree to which a tie functions as a bridge in a network, we can ascertain which ties are more likely to disrupt the flow of resources through a network if they are removed (Freeman 2011; Girvan and Newman 2002). We take up this method in Section 8.6.

Implicit in all of these approaches is that identifying brokers and the ties that bind them reveals the cohesive subgroups of which they are (and are not) a part. Put differently, these approaches bring together aspects of the previous two chapters; Section 8.5 focuses on an algorithm that explicitly brings these two aspects together. It assumes that brokerage is a function of the different groups with which actors are affiliated; thus, not only does it require network data, but it also requires attribute data indicating the specific groups to which people belong.

8.2 Structural Holes

In order to understand Ron Burt's notion of structural holes, we need to briefly return to what Mark Granovetter discovered when exploring how people had acquired their current jobs. He found that they were far more likely to have used personal contacts in finding their present job than by other methods. Moreover, of those who found their jobs through personal contacts, most of these contacts were acquaintances (i.e., weak ties) not close friends (i.e., strong ties). Why? According to Granovetter this occurs because our acquaintances are less likely to be socially involved with one another than are our close friends. Aquaintances play an important role in terms of the overall structure of a network, because they form the crucial bridges that tie together densely knit clusters of people. In fact, if it were not for these weak ties, these clusters would not be connected at all. Granovetter went so far as to argue that, at least in the long run, although not all weak ties are bridges, all bridges are weak ties.

Burt builds on Granovetter's argument, but he takes exception to the idea that only weak ties can be bridges. He concedes that weak ties are more likely to be bridges than are strong ties, but he contends that both can function as bridges. By making this theoretical move, Burt directs attention away from the type of tie a particular bridge is (a construct that is often difficult to measure) and toward the gap in the social structure it spans. Burt refers to these gaps in the social structure as "structural holes" and argues that individuals whose ties span these gaps, regardless of whether they are weak or strong, are at a competitive advantage over those whose ties do not span these gaps, because such ties provide them with the opportunity to broker the flow of various resources through the network. In constructing his structural holes measure, Burt focuses on an actor's ego networks (i.e., the actor, their neighbors, and the ties between them), the triads in which they are embedded, and the constraint (or lack thereof) their position in these triads places on them. To get a sense of this, compare the four different types of triads pictured in Figure 8.1 (adapted from de Nooy et al. 2005:144), which consists of three actors (Ed, Jake, and Isabella) and the ties among them. According to Georg Simmel (1950c), when three people are fully connected (i.e., closed), such as in the first triad, they share norms, create trust, and manage conflicts (de Nooy et al. 2005:144). However, in the other three triads, which are open, the actors in the middle are at an advantage because they are in a position to broker between the other two actors. Moreover, in complete

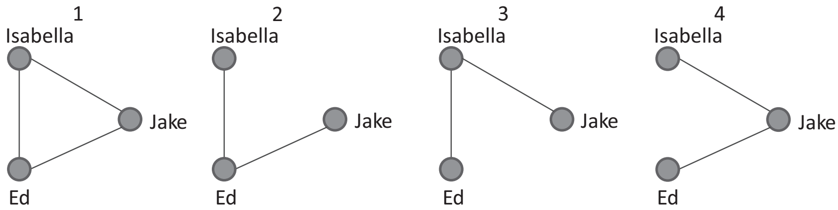


Figure 8.1. Four Types of Triads

triads actors cannot dissolve either tie without putting themselves at a disadvantage. For example, in the first triad, Ed has to maintain his ties to Isabella and Jake if he is to prevent either one from finding themselves in a position of brokerage. For instance, if he were to cut his tie to Jake (the result of which is Triad 3), then Isabella would be in a position of brokerage that she could exploit. Of course, Ed would benefit (and probably prefer) if Isabella cut her tie to Jake (resulting in Triad 2); Jake undoubtedly feels the same way about Isabella's tie to Ed.

In short, open triads provide brokerage opportunities for some actors, whereas complete triads offer only constraints. That is why Burt's structural holes measure does not identify structural holes per se but rather estimates the constraint that all actors in a network face, in light of all the triads in which they are embedded. Less constraint means more autonomy and greater brokerage potential. At this point we do not have to explore the intricacies of how Burt calculates constraint,[1] but the following example (Figure 8.2) adapted from de Nooy et al. (2005:146) should provide a basic understanding of the assumptions lying behind it. First, consider Rick's tie to Victor. It is characterized by three open triads because Guillermo, Enzo, and Renault are not directly connected to Victor. This provides Rick with the opportunity to broker between Victor and Guillermo, Victor and Enzo, and Victor and Renault. By contrast, the constraint placed on Rick because of his ties with Guillermo, Enzo, and Renault is quite high because he is embedded in three complete triads (Rick-Guillermo-Enzo, Rick-Enzo-Renault, and Rick-Guillermo-Renault). Thus, if he were to withdraw from any of these ties, he would place one of them in a position of brokerage. Nevertheless, if Guillermo, Enzo, and Renault have no other ties, they face more constraint (and less autonomy) than Rick because Rick is embedded in three open triads, while Guillermo, Enzo, and Renault are embedded only in closed ones.[2]

Basically, Burt's structural holes algorithm calculates each actor's constraint based on the types of triads in which they are involved, weights

[1] We will explore some of these details as we examine how UCINET and Pajek calculate actor constraint and brokerage potential.

[2] Whether Victor faces more or less constraint than Rick depends on whether he has ties to others who do not share a connection with Rick.

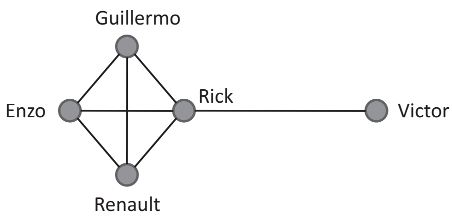


Figure 8.2. Victor's Ego Network (from de Nooy et al. 2005:146)

this by the number of ties in which an actor is involved, and then sums the resulting calculations to arrive at a measure of constraint. If you subtract an actor's constraint score from 1.125 (i.e., the maximum value of constraint – see Buskens and van de Rijt 2008), you have a measure of autonomy or brokerage potential. In keeping with the pattern of the previous chapters, we first examine how UCINET and NetDraw implement Burt's measure before turning to Pajek and ORA. All four programs calculate the aggregate constraint for each actor, but only NetDraw provides an additive inverse of the constraint measure (i.e., a measure of autonomy), which is helpful when visualizing network maps where node size varies by constraint/autonomy, as it is somewhat more intuitive to vary the node size where larger node size indicates more autonomy and less constraint. We will also see how Burt's measures assign isolates the lowest possible level of constraint (and thus high brokerage potential), which, of course, makes no sense because it is hard to imagine how an actor with no ties is in a position to broker anything. In UCINET, NetDraw, and ORA we have to adjust for this (or at least take it into account), while in Pajek we do not because it makes the adjustment on its own.

Constraint (Structural Holes) in UCINET and NetDraw

[UCINET] Network>Ego Networks >Structural Holes

Using Noordin's alive communication network (`Alive Communication Network.##h`), select UCINET's *Network>Ego Networks >Structural Holes* command. In the resulting dialog box (Figure 8.3),

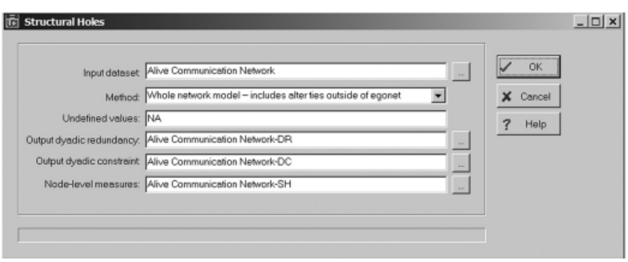
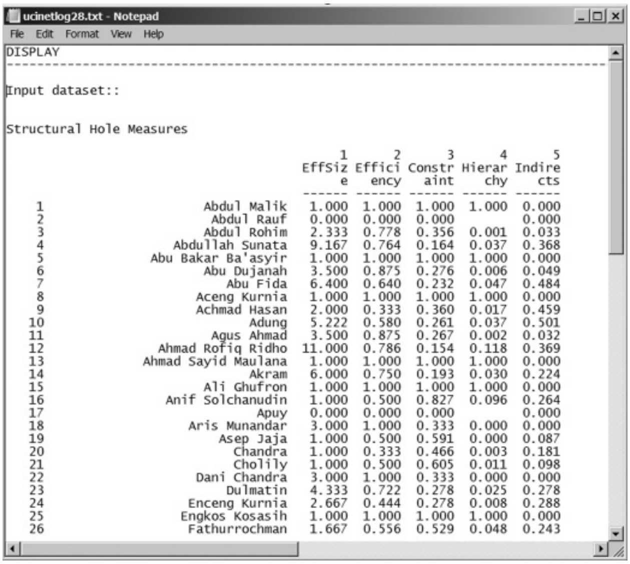


Figure 8.3. UCINET's Structural Holes Dialog Box



ucinetlog28.txt - Notepad

File Edit Format View Help

DISPLAY

Input dataset::

Structural Hole Measures

		1 EffSize	2 Efficiency	3 Constraint	4 Hierarchy	5 Indirects
1	Abdul Malik	1.000	1.000	1.000	1.000	0.000
2	Abdul Rauf	0.000	0.000	0.000		0.000
3	Abdul Rohim	2.333	0.778	0.356	0.001	0.033
4	Abdullah Sunata	9.167	0.764	0.164	0.037	0.368
5	Abu Bakar Ba'asyir	1.000	1.000	1.000	1.000	0.000
6	Abu Dujanah	3.500	0.875	0.276	0.006	0.049
7	Abu Fida	6.400	0.640	0.232	0.047	0.484
8	Aceng Kurnia	1.000	1.000	1.000	1.000	0.000
9	Achmad Hasan	2.000	0.333	0.360	0.017	0.459
10	Adung	5.222	0.580	0.261	0.037	0.501
11	Agus Ahmad	3.500	0.875	0.267	0.002	0.032
12	Ahmad Rofiq Ridho	11.000	0.786	0.154	0.118	0.369
13	Ahmad Sayid Maulana	1.000	1.000	1.000	1.000	0.000
14	Akram	6.000	0.750	0.193	0.030	0.224
15	Ali Ghufron	1.000	1.000	1.000	1.000	0.000
16	Anif Solchanudin	1.000	0.500	0.827	0.096	0.264
17	Apuy	0.000	0.000	0.000		0.000
18	Aris Munandar	3.000	1.000	0.333	0.000	0.000
19	Asep Jaja	1.000	0.500	0.591	0.000	0.087
20	Chandra	1.000	0.333	0.466	0.003	0.181
21	Cholily	1.000	0.500	0.605	0.011	0.098
22	Dani Chandra	3.000	1.000	0.333	0.000	0.000
23	Dulmatin	4.333	0.722	0.278	0.025	0.278
24	Enceng Kurnia	2.667	0.444	0.278	0.008	0.288
25	Engkos Kosasih	1.000	1.000	1.000	1.000	0.000
26	Fathurrochman	1.667	0.556	0.529	0.048	0.243

Figure 8.4. UCINET's Structural Holes Output

select the "Whole network model – includes alter ties outside of egonet" option and click "OK."

The output log, a portion of which is displayed in Figure 8.4, first lists measures of dyadic redundancy and dyadic constraint, which are used in calculating aggregate constraint, although the details of their calculation need not concern us here.[3] Next, it lists Burt's constraint measure along with several other related measures. Note that Abdul Rauf's and Apuy's constraint scores equal 0.00 even though they are isolates. They (and all other isolates) should be changed to 1.125. The easiest way to do this in UCINET (at least with a network of this size) is to open the network data in UCINET's spreadsheet editor and make the change yourself.

Data>Data Editors >Matrix Editor

What might we expect to find from an analysis of a network or series of related networks using Burt's measure? Given what we have seen in previous chapters, we might suppose that there will be a positive correlation of constraint scores across networks. We would also probably not be too surprised if a negative correlation existed between Burt's measure of constraint and betweenness centrality, as both attempt to tap into the brokerage potential of actors in a network. Table 8.1 summarizes the correlation of constraint scores with those of other networks (first five rows) and with the betweenness scores of their corresponding network (sixth row). Clearly, a correlation of constraint scores does exist

[3] Hanneman and Riddle (2005:Chapter 9) provide a nice summary of all of UCINET's output.

Table 8.1. *Correlation of constraint and betweenness scores*

	Trust network constraint	Operational network constraint	Communi-cation network constraint	Business & finance network constraint	Combined network constraint
Trust Constraint	1.000	0.312	0.516	0.218	0.506
Operational Constraint		1.000	0.329	0.138	0.798
Communication Constraint			1.000	0.189	0.474
Business & Finance Constraint				1.000	0.149
Combined Constraint					1.000
Betweenness Centrality	−0.551	−0.287	−0.369	−0.287	−0.263

across networks although it is stronger between some than others. For example, the degree of constraint within the trust and communication networks is strongly correlated (0.516), which may indicate that those who broker the flow of communication through the network are also those who lie in positions of brokerage within the trust network. By contrast, but not surprisingly, the correlations between the constraint scores of the business and finance network and those of the other networks are far weaker, indicating once again that the business and finance network differs substantially from the other networks. Comparing the constraint scores of each network to the betweenness scores of the corresponding network, we can see that in all cases a negative correlation exists. Moreover, all of the correlations are relatively strong, which probably indicates that Burt's measure of constraint is capturing much of the same dimension of brokerage as is betweenness. That said, the correlations are not perfect (i.e., −1.00), which tells us that, whereas betweenness helped identify individuals within these networks who were in positions of brokerage, it appears that Burt's measure of constraint has identified others whom we should identify and take note of.

[NetDraw] File >Open>Ucinet dataset >Networks

[NetDraw] File >Open>Ucinet dataset >Attribute data

Because network visualization should be a regular companion of the estimation of metrics (Brandes, Raab, and Wagner 2001), open the alive communication network (i.e., `Alive Communication Network.##h`) in NetDraw. Also read in (as attribute data) the data file containing the structural holes measures generated by UCINET (probably named `Alive Communication Network-SH`) using NetDraw's *File>Open>Ucinet dataset>Attribute data* command. Then vary the size

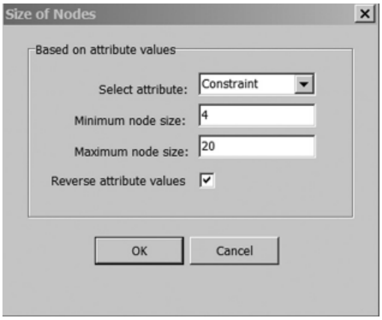


Figure 8.5. NetDraw's Size of Nodes Dialog Box

Properties >Nodes >Symbols>Size >Attribute-based

of the nodes to reflect the constraint of each actor using *Properties> Nodes>Symbols>Size>Attribute-based* command. In the dialog box that this command calls up, select the "Constraint" attribute and check the "Reverse attribute values" box (see Figure 8.5); this tells NetDraw to draw the size of the nodes in terms of the inverse of Burt's measure of constraint. This should produce a network map similar to Figure 8.6. Not surprisingly, Burt's measure has also found that within the communication network Noordin Top is in a position of brokerage, as are a number of the individuals identified in our previous analysis of the communication network using betweenness centrality (Table 7.3). However, the variation in Burt's measure of constraint is far less than the variation in betweenness centrality scores, which may indicate that brokerage potential is more widespread in the network than previously thought.

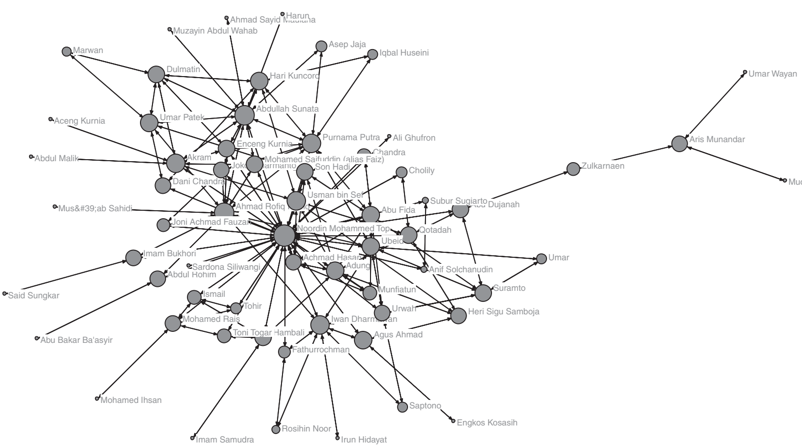


Figure 8.6. NetDraw Map of the Strike Network's Structural Holes (rConstraint)

Analysis> Structural Holes> Whole Network model

NetDraw also provides for the calculation of structural holes as well with its *Analysis>Structural Holes>Whole Network model* command. It also generates an *rConstraint* attribute (i.e., the additive inverse of *Constraint*) that you can use to visualize your network without checking the "Reverse attribute values" box in the "Size of Nodes" dialog box as we did previously.

Constraint (Structural Holes) in Pajek

[Pajek Main Screen] Net> Vector >Structural Holes

Read the alive Noordin network project file (`Alive Noordin Network.paj`) into Pajek. In Pajek the command *Net>Vector>Structural Holes* computes the structural holes measures for all the actors in a network as well as two new networks, proportional strength and dyadic constraint, in which the line values equal the strength and constraint between actors. If you draw the dyadic constraint network, actors with relations characterized by high constraint will be drawn closer together than they would in the original network. Similarly, those with low constraint will be drawn farther apart, which, in turn, should create a space between them that looks something like a hole or gap in the social structure.[4] Figure 8.7 attempts to capture the difference between the two networks. The lower panel is the alive communication network, in which the ties between actors equal the level of dyadic constraint between them; the upper panel is the original alive communication network in which the ties are simply the communication ties between actors. Comparing the two networks you can see that in the lower panel the gap between Noordin and the rest of those to whom he has ties is much larger than in the upper panel, not surprisingly indicating that Noordin sits aside a structural hole and is in a position of brokerage. Compare also the gap between Hari Kuncoro and Dulmatin in the upper-right portion of the network maps. It is relatively small in the regular network (upper panel), but quite pronounced in the dyadic constraint network (lower panel). This suggests that Kuncoro and Dulmatin are sitting aside a structural hole and able to control some of the communication that passes through the network. Thus, dissolving the tie between them could possibly disrupt the ability of members of Noordin's network to communicate with one another. Of course, Kuncoro's and Dulmatin's peripheral positions in the network could mitigate the effect.

[Pajek Main Screen] Draw>Draw-Vector

If you want the size of the actors in your drawing to represent their respective aggregate constraint, then use Pajek's *Draw>Draw-Vector* command, making sure that aggregate constraint vector is showing in the first Vector drop-down box. At the draw screen, you may want to

[4] UCINET also generates a dyadic constraint matrix or network that can be visualized in this way in NetDraw.

Figure 8.7. Alive Communication Network (Pajek): Original and Dyadic Constraint

select the *Autosize* option in the *Options>Size>of Vertices* submenu; otherwise, the vertices will be drawn too small. If they are still too small after choosing the auto-size option, try setting the size of the vertices to a larger size, such as "8." If you want the size of the vertex to be positively related to the inverse of the aggregate constraint (i.e., you want larger vertices to indicate better brokerage potential), select the *Vector>Transform>Add Constant* command, and in the dialog box type in the value "–1.000."[5] Then redraw the network as before, except with the newly created vector showing in the Vector drop-down menu. Pajek will inform you that it has drawn negative lines as positive lines, which is OK because all the lines are negative. You should get a network map that looks similar to Figure 8.8.

[Pajek Draw Screen] Options>Size >of Vertices

[Pajek Main Screen] Vector >Transform >Add Constant

[5] Pajek assigns a maximum value of 1.00 when calculating constraint, so rather than adding −1.125 here, we add only −1.00.

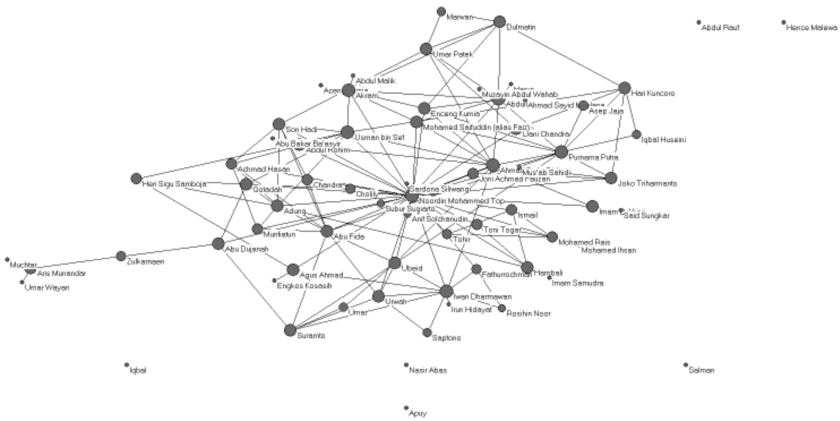


Figure 8.8. Pajek Network Map of Communication Network's Structural Holes

As in Figure 8.6, Noordin appears to be the actor with the most brokerage potential in the network but his brokerage "advantage" is not substantial given the lack of variation in the network.

Constraint (Structural Holes) in ORA

[ORA-Main Screen] Analysis >Generate Reports >Show me everything (All Measures)

Open the alive Noordin meta-network (`Alive Noordin Network .xml`) into ORA. Currently, in order to calculate constraint in ORA analysts have to select ORA's "All Measures" report: *Analysis>Generate Reports>Show me everything (All Measures)*. As with most of ORA's reports, you will need to work through a number of dialog boxes (not shown) before the report is generated. Typically, you can accept ORA's default settings. Interestingly, with this report ORA reports the results for all the actors in the network rather than those ranked in the top ten, which it typically reports (you can change this if you prefer in the third dialog box). If you plan on merging the results with other data, then you may want to select the CSV option in the final dialog box; this will generate a spreadsheet-like output than be easily imported into spreadsheet programs such as Excel. Finally, because this is an all measure report, it can take some time to be generated.

Visualizations >View Networks >2-D Visualization

If you do not wish to generate a report with the structural holes measures for all the actors in a network but instead want to visualize it such that the sizes of the nodes vary in terms of Burt's measure of constraint, first visualize the communication network using ORA's *Visualizations>View Networks>2-D Visualization* command (or its companion speed button on the Editor panel). Next at ORA's visualizer, vary the size of each node by Burt's measure of constraint with ORA's

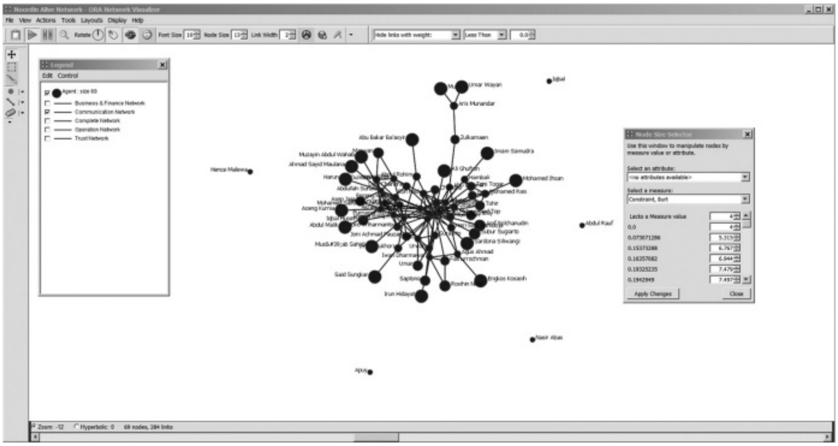


Figure 8.9. ORA Network Map of Communication Network's Structural Holes

Display>Node Appearance>Size Nodes by Attribute or Measure command. In the "Select a Measure" option in the dialog box that is called up, select "Constraint, Burt," and you should get a network map similar to Figure 8.9. Note that unlike the previous graphs we have generated, here larger nodes indicate higher levels of constraint and, concomitantly, lower brokerage potential. Note also, however, that ORA does not adjust constraint scores if an actor is an isolate. The node size of the isolates in the graph indicates low constraint and high brokerage potential, which, as we discussed previously, makes little sense.

[ORA-Visualizer] Display>Node Appearance >Size Nodes by Attribute or Measure

One feature of ORA that is quite useful for analyzing and comparing metrics is its "Measure Charts" function, which is accessed either through the *Visualizations>Measure Charts* command or the companion speed button on ORA's Editor panel. After selecting the Noordin alive network in the Meta-Network Manager and issuing the command, a dialog box appears (not shown here). Accept ORA's defaults and click "Next." This brings up ORA's measure charts interface (Figure 8.10), which most likely will default to the Bar Charts tab; although, as you can see, several options are available (scatter plots, histograms, regression). You will probably want to explore all of these at some point, but here we will focus on the scatter plot feature. Click on the Scatter Plot tab. For the X-Axis measure, indicate you want the betweenness centrality (i.e., "Centrality, Betweenness") scores of the communication network. For the Y-Axis measure, indicate you want the Burt's measure of constraint (i.e., "Constraint, Burt") scores of the communication network. This will produce a scatter plot similar to Figure 8.10. As expected, there is a negative association between the two metrics (as indicated by the downward sloping line). ORA includes a few metrics with the scatter

[ORA-Measure Charts] Visualizations >Measure Charts

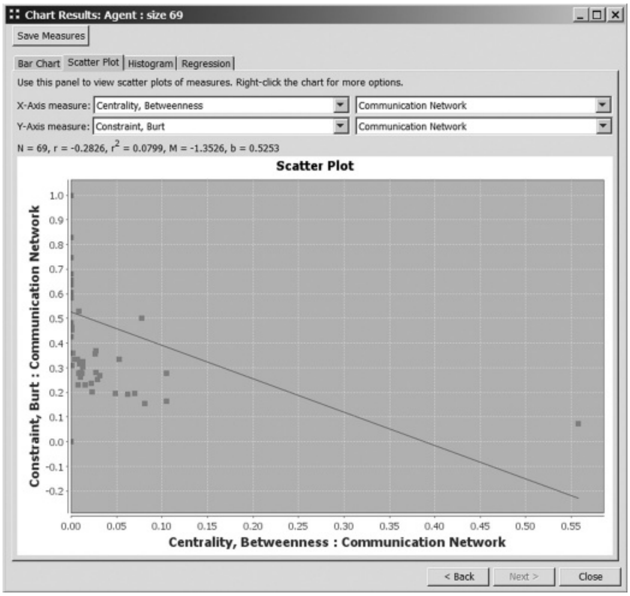


Figure 8.10. Scatter Plot Comparison of Constraint and Betweenness (ORA)

plot, including the correlation between the two measures: $r = -.2826$. However, if you compare this to the correlation we got with UCINET (Table 8.1), you will note that they are not the same. The reason is simple: We have already seen that ORA does not adjust the constraint score of isolates to reflect the inability of isolates to broker anything. If you recall, UCINET does not either (Pajek does, however), but we did adjust the scores manually before estimating the correlation between the two metrics. Thus, while this is a useful feature, it should be used with some caution.

8.3 Bridges, Bi-Components, and Cutpoints

In order to illustrate how the location of certain ties and actors can be crucial to the flow of material and nonmaterial resources through the network, let's turn to Noordin's alive and free operational network (Figure 8.11). Clearly, Noordin, Saptono, and Suramto hold crucial positions in the network because the removal of any one of them (or the dissolution of two of the three ties between them) disconnects the network into separate components. Actors whose removal disconnects the network or disconnects a component of a network are called cutpoints (UCINET and

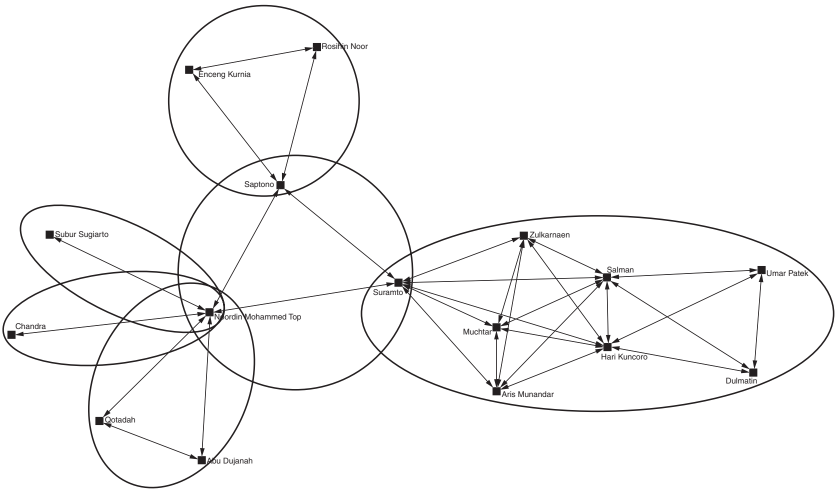


Figure 8.11. Alive and Free Operational Network: Bi-Component Analysis

NetDraw), cut-vertices (Pajek), articulation points (UCINET and Pajek), or boundary spanners (ORA).

Parts of a network that are invulnerable to the removal or isolation of a single actor are known as bi-components. Formally, *bi-components* are components without a cutpoint. They are the sections of a network in which the removal of a single actor does not create a new component. This means that in a bi-component no actor completely controls the flow of resources between two other actors because there is always an alternative path through which resources can flow. Thus, bi-components tend to be more cohesive than strong or weak components because there are at least two different paths between each pair of actors. In the alive and free operational network, there are six bi-components, all of which are circled in Figure 8.11. Often, analysts define a bi-component as a component *of minimum size 3* without a cutpoint, but bi-components of size 2 are of special interest for analysts because they represent ties in the network that if removed or dissolved will disconnect the network. In other words, a bi-component of size 2 is a bridge in the network. In the alive and free operational network there are two such ties (or bridges): the tie between Noordin and Chandra and the tie between Noordin and Sugiarto. Note also that bi-components sometimes overlap with one another. Indeed, where they overlap is where a network's cutpoints are located. Put differently, cutpoints are nodes that belong to two or more bi-components.

Bridges, Bi-Components, and Cutpoints in NetDraw and UCINET

[UCINET] Network >Regions >Bi-Component

In UCINET we detect cutpoints and bi-components using UCINET's *Network>Regions>Bi-Component* command. UCINET does not provide an option for choosing a minimum size of bi-components; instead, it detects bi-components of all sizes (Figure 8.12). Applying this command to the alive and free operational network (`Alive & Free Operational Network.##h`), UCINET's output log first indicates the members of each block (i.e., bi-component) and then lists the articulation points (i.e., cutpoints) in the network. A quick glance indicates that it agrees with what is illustrated in Figure 8.11.

UCINET also generates a partition (Cutpoints.##h) that can be used to highlight the cutpoints in NetDraw. However, because NetDraw's bi-component routine works quite well and is superior to UCINET's in some respects, there is no need to import the partition files created in UCINET into NetDraw. Instead, use NetDraw's *Analysis>Blocks & Cutpoints* command, which not only automatically assigns the cutpoints one color and the non-cutpoints another, but also creates partitions that can be used to highlight the various bi-components in the network. Indeed, it was this combination of features that helped generate the network map in Figure 8.11.[6]

[NetDraw] Analysis >Blocks & Cutpoints

[NetDraw] Properties >Nodes> Symbols>Color >Attribute-based

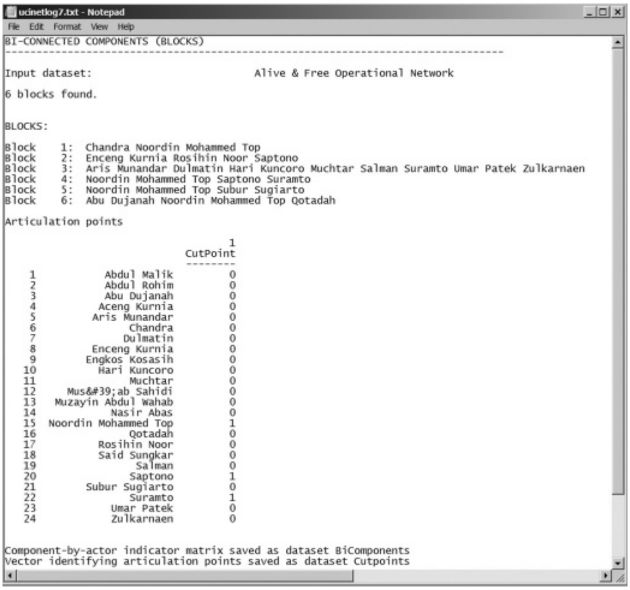
If you want to change the color of the cutpoints or color the nodes to reflect the bi-component of which they are a part, use the *Properties>Nodes>Symbols>Color>Attribute-based* command. In the resulting dialog box, select the block (i.e., bi-component) you want to highlight (Figure 8.13).

Bridges, Bi-Components, and Cutpoints in Pajek

[Pajek] Net >Components >Bi-Components

Read Noordin's alive and free project file (`Alive and Free Noordin Network.paj`) into Pajek. In Pajek, the *Bi-components* command is found under the *Net>Components* submenu. Upon selecting this command, a dialog box prompts you to specify the minimum size of the bi-components to be identified. While the default value (three) will identify the bi-components within the network, it will only report cutpoints that connect two or more bi-components of size 3. If you select a minimum size of two, it will identify all bi-components, bridges, and cutpoints, including cutpoints connecting bridges. Pajek will ask if you want to overwrite the current network with a partition of lines. Select "No" (it is unclear why "Yes" is the default), click "OK," and the *Bi-components* command will generate a new network, two partitions, and something we have not

6 Whereas NetDraw identified the various bi-components in the network, the drawing tool in Microsoft Word was used to circle the bi-components in Figure 8.11.



```
ucinetlog7.txt - Notepad
File Edit Format View Help
BI-CONNECTED COMPONENTS (BLOCKS)
--------------------------------------------------------------------------------

Input dataset:                        Alive & Free Operational Network

6 blocks found.

BLOCKS:

Block   1:  Chandra Noordin Mohammed Top
Block   2:  Enceng Kurnia Rosihin Noor Saptono
Block   3:  Aris Munandar Dulmatin Hari Kuncoro Muchtar Salman Suramto Umar Patek Zulkarnaen
Block   4:  Noordin Mohammed Top Saptono Suramto
Block   5:  Noordin Mohammed Top Subur Sugiarto
Block   6:  Abu Dujanah Noordin Mohammed Top Qotadah

Articulation points

                                   1
                            CutPoint
                            --------
  1          Abdul Malik           0
  2          Abdul Rohim           0
  3          Abu Dujanah           0
  4         Aceng Kurnia           0
  5        Aris Munandar           0
  6              Chandra           0
  7             Dulmatin           0
  8        Enceng Kurnia           0
  9       Engkos Kosasih           0
 10         Hari Kuncoro           0
 11              Muchtar           0
 12      Mus&#39;ab Sahidi           0
 13   Muzayin Abdul Wahab           0
 14          Nasir Abas           0
 15  Noordin Mohammed Top           1
 16              Qotadah           0
 17         Rosihin Noor           0
 18         Said Sungkar           0
 19               Salman           0
 20              Saptono           1
 21       Subur Sugiarto           0
 22              Suramto           1
 23           Umar Patek           0
 24           Zulkarnaen           0


Component-by-actor indicator matrix saved as dataset BiComponents
Vector identifying articulation points saved as dataset Cutpoints
```

Figure 8.12. UCINET Bi-Component Output Log

encountered until now: a hierarchy. If you draw the new network and in the draw screen select the *Options>Colors>Edges>Relation Number* command, the lines will be different colors, reflecting the bi-components of which they are a part. The first partition ("Vertices belonging to exactly one bi-component") indicates the bi-component class (i.e., number) to which an actor belongs. Actors that do not belong to a bi-component (e.g., isolates) are assigned to class 0, and actors belonging to two or more bi-components (i.e., cutpoints) are assigned to class 9999998.

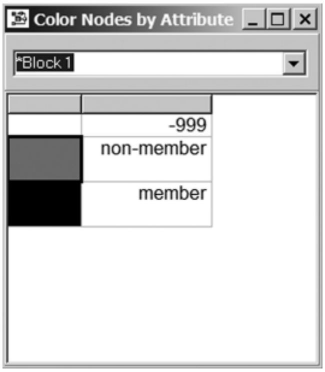


Figure 8.13. NetDraw Color of Nodes Dialog Box

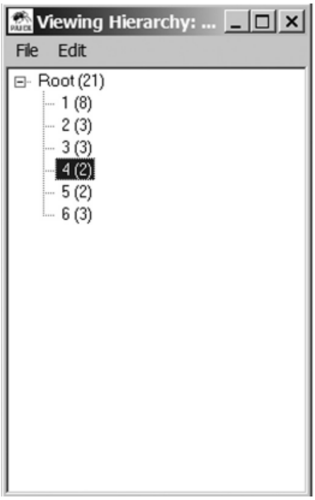


Figure 8.14. Pajek Bi-Component Hierarchy

The second partition ("Articulation points") indicates the number of bi-components to which a vertex belongs: 0 for isolates, 1 for actors that belong to exactly one bi-component, 2 for actors that belong to two bi-components, and so on. Finally, the hierarchy shows the bi-components to which each actor belongs. Pajek uses hierarchy objects to store the bi-components because cutpoints can belong to two or more bi-components.

Because bridges are components of size 2 in an undirected network without multiple lines, it is easy to find the bridges in the hierarchy of bi-components: Open the Edit screen with the hierarchy of bi-components with the command *File>Hierarchy>Edit* or with the Edit button on the left of the Hierarchy drop-down menu. Next, click on the "+" sign to the left of the word "Root." This should produce a figure similar to Figure 8.14; this lists the six bi-components in the alive and free operational network. The size of each subnetwork is reported between parentheses, as to identify the actors that are a part of the two bi-components of size 2.

File>Hierarchy >Edit

Draw>Draw-Partition

Next, open the draw screen using the *Draw>Draw-Partition* option from the main menu, making sure that the partition "Articulation points" is highlighted at the first Partition drop-down menu. Now, you should see a drawing (Figure 8.15), where the isolates are one color (e.g., white), most of the members of the operational network are another color (e.g., black), and a handful of employees are yet another color (e.g., various shades of gray). Here, the actors that are not black or white are cutpoints in the network.

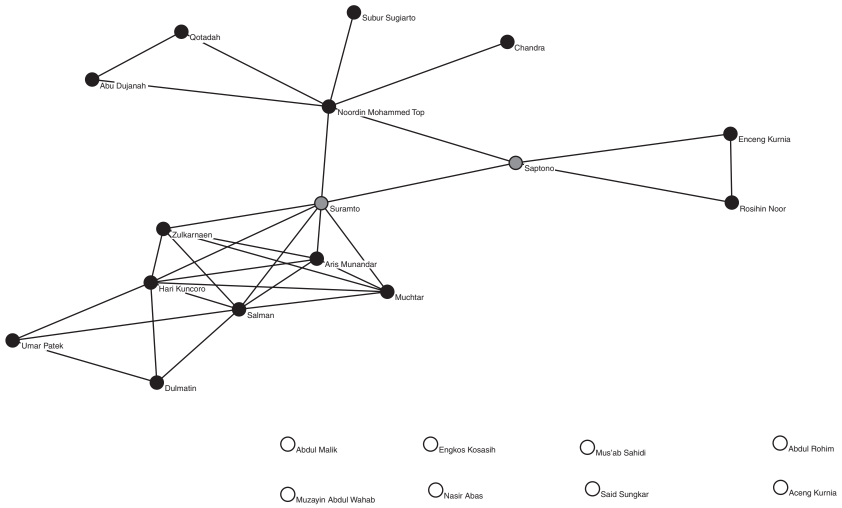


Figure 8.15. Alive and Free Operational Network with Cutpoints Highlighted (Pajek)

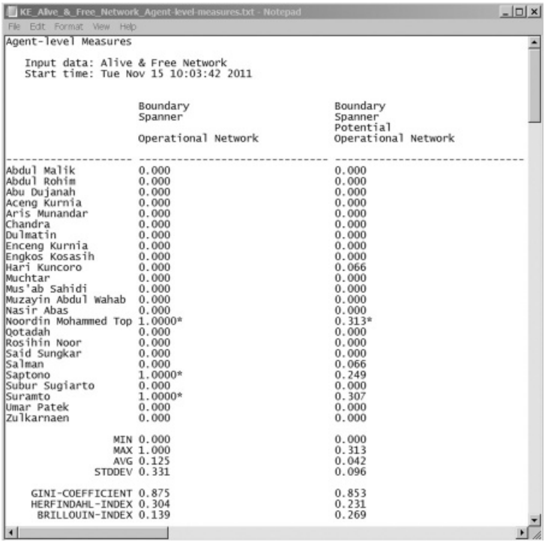
Cutpoints (Boundary Spanners) in ORA

Open Noordin's alive and free meta-network (`Alive and Free Noordin Network.xml`) into ORA. Currently, ORA does not detect bi-components; it does, however, identify cutpoints, which it calls *boundary spanners*. One way that you can identify cutpoints in ORA is through the "Show me everything report," which we previously referenced in the discussion of constraint and structural holes. This will generate a report, a portion of which will contain information similar to that presented in Figure 8.16. Note that cutpoints are assigned a score of 1.00, and as one would expect (and hope) these results mirror those that we found using UCINET, NetDraw, and Pajek. A related metric that may also be of interest to analysts, is generated by the same report, and immediately follows the cutpoint results is what ORA refers to as *Boundary Spanner, Potential* (Cormen et al. 2009), which is the ratio of betweenness centrality to degree centrality. It assumes that actors that score high in terms of the former but low in terms of the latter potentially act as ties between groups of entities.

[ORA-Main Screen] Analysis >General Reports>Show me everything (All Measures)

In addition to the all-measures report, analysts can identify boundary spanners using the ORA visualizer or its *Measure Charts* function. For example, Figure 8.17 presents the bar chart generated by the measure charts function. Note that ORA allows you to control the number of actors (nodes) that are displayed, how you want to sort the results (by value or name), and whether you want the names of the actors displayed.

Visualizations >View Networks >2D Visualization


KE_Alive_&_Free_Network_Agent-level-measures.txt - Notepad

File Edit Format View Help

Agent-level Measures

Input data: Alive & Free Network
Start time: Tue Nov 15 10:03:42 2011

	Boundary Spanner Operational Network	Boundary Spanner Potential Operational Network
Abdul Malik	0.000	0.000
Abdul Rohim	0.000	0.000
Abu Dujanah	0.000	0.000
Aceng Kurnia	0.000	0.000
Aris Munandar	0.000	0.000
Chandra	0.000	0.000
Dulmatin	0.000	0.000
Enceng Kurnia	0.000	0.000
Engkos Kosasih	0.000	0.000
Hari Kuncoro	0.000	0.066
Muchtar	0.000	0.000
Mus'ab Sahidi	0.000	0.000
Muzayin Abdul Wahab	0.000	0.000
Nasir Abas	0.000	0.000
Noordin Mohammed Top	1.0000*	0.313*
Qotadah	0.000	0.000
Rosihin Noor	0.000	0.000
Said Sungkar	0.000	0.000
Salman	0.000	0.066
Saptono	1.0000*	0.249
Subur Sugiarto	0.000	0.000
Suramto	1.0000*	0.307
Umar Patek	0.000	0.000
Zulkarnaen	0.000	0.000
MIN	0.000	0.000
MAX	1.000	0.313
AVG	0.125	0.042
STDDEV	0.331	0.096
GINI-COEFFICIENT	0.875	0.853
HERFINDAHL-INDEX	0.304	0.231
BRILLOUIN-INDEX	0.139	0.269

Figure 8.16. ORA All Measures Report Identifying Boundary Spanners (Cutpoints)

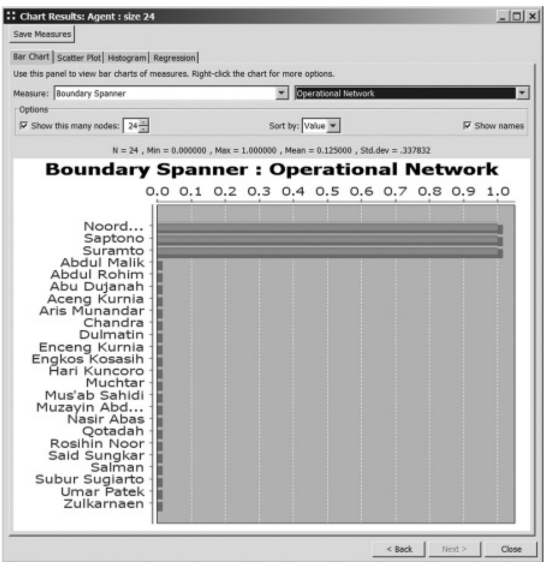


Figure 8.17. ORA Measure Charts Function Identifying Boundary Spanners

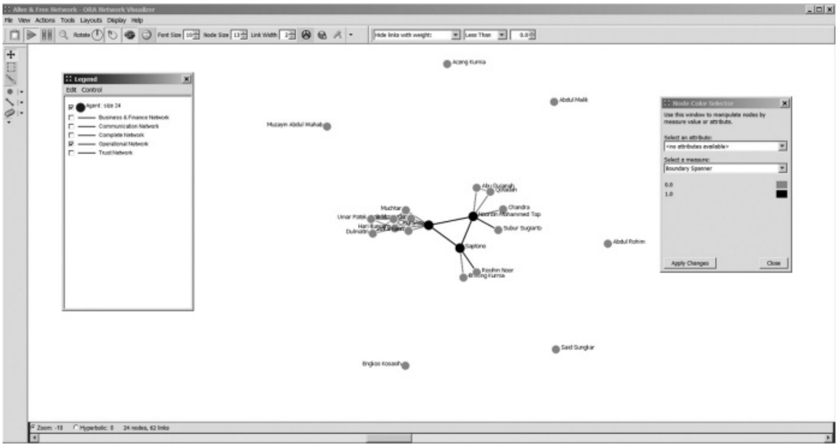


Figure 8.18. Boundary Spanners Identified in ORA's Visualizer

The default number of displayed nodes is ten, but here we have raised it to twenty-four so that results for all of the actors in the network are displayed. Obviously, displaying the results for all network members is only realistic with small networks like this one. When working with large networks, you will want to try alternative numbers of nodes until you find one that adequately captures the results. As you may have already discovered or guessed, ORA's bar chart function allows you to display the results for all of the metrics estimated by ORA (e.g., boundary spanner potential, various centrality measures, constraint, and so on).

Visualization >View Networks >Measure Charts

Finally, probably the best approach for using ORA's visualizer to identify the cutpoints in a network is by adjusting the color (rather than the size) of the nodes. In other words, using the *Display>Node Appearance>Node Color>Color Nodes by Attribute or Measure* command, adjust the color of the nodes (see Figure 8.18) to indicate whether a particular actor is a boundary spanner or not. As you can see from the resulting network map, it has produced results similar to those displayed in Figures 8.11 and 8.15.

[ORA Visualizer] Display >Node Appearance >Node Color >Color Nodes by Attribute or Measure

8.4 Key Players

While intuitively appealing, identifying cutpoints is somewhat limited, as there has to exist a single actor that disconnects a network. In the real world, this can be a rare occurrence because most networks are too well connected. Instead, what we need is an algorithm that can identify an optimal set of actors that either completely disconnects the network or at least fragments it to such an extent that it makes the flow of resources across the network (e.g., communication) more difficult. To complicate matters somewhat, simply removing, say, the five or ten most central actors in a network (however, one defines "central") will not necessarily do the trick either, because highly central actors often reach or connect the same actors and groups. To address this issue Borgatti (2006) developed an algorithm that seeks to identify an optimal *set of actors* whose removal either *disconnects or significantly fragments* the network. Two variations of the algorithm exist. The first ("Fragmentation") uses the standard measure of fragmentation (discussed in Chapter 5) to gauge how much various sets of actors fragment the network when they are removed from the network. That is, a fragmentation score is calculated both prior to and after the removal of each of the sets, and the set that increases the level of fragmentation the most is considered optimal. The second variation ("Distance-weighted Fragmentation") is similar to the first except that rather than using the standard fragmentation measure, it uses a distance-weighted measure (also discussed in Chapter 5), that essentially identifies the optimal set of actors whose removal lengthens the average distance (in terms of path length) between all pairs of actors in the network.

Recognizing that the removal of actors may not always be the best or desired strategy when working with dark networks, and that analysts may want to "select an efficient set of actors to surveil, to turn (as into double-agents), or to feed misinformation to," Borgatti has developed an additional algorithm that looks "for a set of network nodes that are optimally positioned to quickly diffuse information, attitudes, behaviors or goods and/or quickly receive the same (Borgatti 2006:22). Put differently, this algorithm is designed to find the optimal set of actors that reaches the highest number of other actors. Here again, Borgatti has developed two variations on this algorithm. The first ("Percent Nodes Reached") simply counts the proportion of distinct actors reached by the set of key actors, while the second ("Distance-weighted Reach") weights this calculation by the path distance between the set of key actors and all other actors in the network.[7]

While it has yet to be implemented in UCINET, it has (somewhat) been implemented in NetDraw. Better than NetDraw's version, however, is Borgatti's *Key Player* program (2011), which comes with each version of UCINET. ORA has also implemented the key player algorithms (which it calls "critical sets") although it currently does not include the distance-weighted versions of the algorithms. We begin by examining how Borgatti's Key Player program has implemented these algorithms before turning to see how ORA has done the same.

[7] Both distance-weighted algorithms use average reciprocal distance (ARD) in their calculations rather than the standard measure of closeness because (as we discussed in the previous chapter) the former can be used with disconnected graphs while the latter cannot.

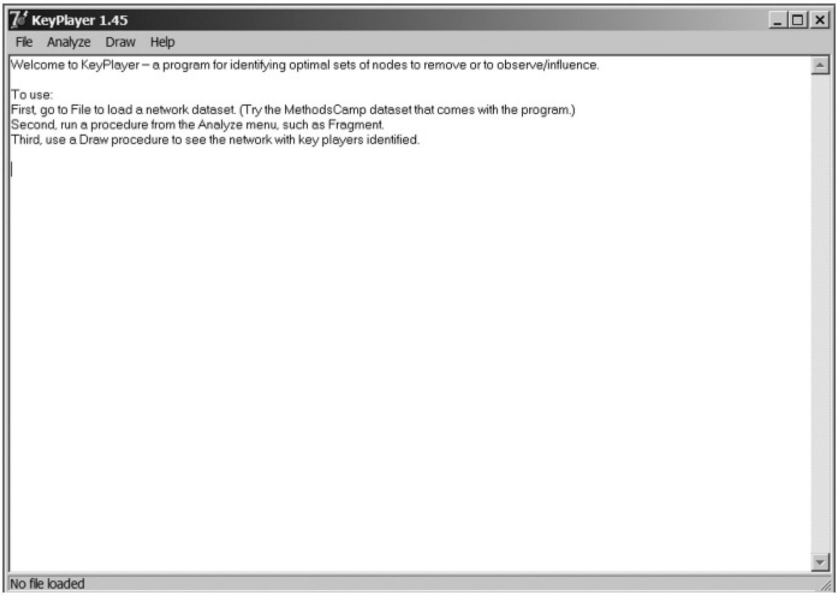


Figure 8.19. Key Player Program

Identifying Key Players with Key Player

[Key Player] File>Load Network Data >UCINET Dataset

Analyze >KPP-1 Select Nodes to Remove >Fragmentation Criterion

Borgatti's Key Player program can be found in the *Analytic Technologies >Helper Applications>Key Player* folder in the *Program Files* directory. It reads UCINET files and is relatively straightforward to use. Open Key Player by double-clicking on the icon in the Key Player folder, which will open an interface that looks similar to Figure 8.19. Load the combined alive network (`Alive Combined Network.##h`) using the *File>Load Network Data>UCINET Dataset* command. To run the standard key player fragmentation algorithm, select the *Analyze>KPP-1 Select Nodes to Remove>Fragmentation Criterion* command. This calls up a dialog box that prompts you to indicate how large you want your key player "set" to be. After indicating the size, click "OK," and the program identifies a set of key players for removal and creates a corresponding partition (`fragmentvec.##h`) that can be inspected or used to visualize in NetDraw. Key Player also provides "before" and "after" fragmentation scores, which can be useful when comparing and choosing between different size sets. For example, if a set of size 6 does not increase fragmentation to any great degree over a set of size 5, then it probably makes more sense to choose the smaller set over the larger one. It is important to note that because Key Player identifies the optimal set, it does not list the actors of the set in rank order. Rather, it lists them in the order that they appear in the dataset. To obtain a set of key

Table 8.2. *Key players in alive combined network*

Fragmentation	Distance-weighted fragmentation	Reach	Distance-weighted reach
Abdullah Sunata	Abdullah Sunata	Abdullah Sunata	Abdullah Sunata
Akram	Akram		Akram
			Imam Samudra
Iwan Dharmawan	Iwan Dharmawan		Iwan Dharmawan
		Mohamed Saifuddin	
		Nasir Abas	
Noordin Top	Noordin Top	Noordin Top	Noordin Top
Rosihin Noor			
	Suramto		
		Usman bin Sef	

Analyze >KPP-1 Select Nodes to Remove >Distance-Weighted Fragmentation Criterion

[NetDraw] File>Open >UCINET Dataset >Network

File>Open >UCINET Dataset >Attribute Data

[Key Player] Analyze >KPP-2 Select Nodes to Utilize ># of Nodes Reached Criterion

Analyze>KPP-2 Select Nodes to Utilize>Distance-Weighted Reach Criterion

players using the distance-weighted algorithm use the *Analyze>KPP-1 Select Nodes to Remove>Distance-Weighted Fragmentation Criterion* command. This generates a corresponding partition (`distvec.##h`) that can be inspected or used for visualization purposes in NetDraw.

To visualize the results, first read the network that you analyzed using Key Player (e.g., `Alive Combined Network.##h`) into NetDraw, and then load the partition generated by Key Player (e.g. `distvec.##h`) as an attribute file. You can then color the nodes, as illustrated previously, and generate a network map similar to Figure 8.20. In this example, five individuals (colored gray) in the alive combined network were identified using the distance-weighted algorithm, three of whom (Noordin, Akram, and Abdulah Sunata) were also cutpoints. The size of a key player set should, of course, take into account relevant factors, such as time constraints and available resources.

The diffusion or "reach" algorithms are accessed and estimated in similar ways to the fragmentation algorithms. To run the standard reach algorithm, use the *Analyze>KPP-2 Select Nodes to Utilize># of Nodes Reached Criterion* command; to run the distance-weighted reach algorithm, use the the *Analyze>KPP-2 Select Nodes to Utilize>Distance-Weighted Reach Criterion* command. As with the fragmentation algorithms, both generate partitions that can be used for visualization or inspection. Unfortunately, both are named `reachvec.##h`, which means that the results of one will overwrite the results of the other if you are not careful. The two algorithms also produce "before" and "after" scores that indicate the effective reach of the set.

Table 8.2 presents the results from a key player analysis of the alive combined network using the four key player algorithms. While they produce similar results, the results differ. The two fragmentation algorithms identified four of the same individuals. They only differ in terms

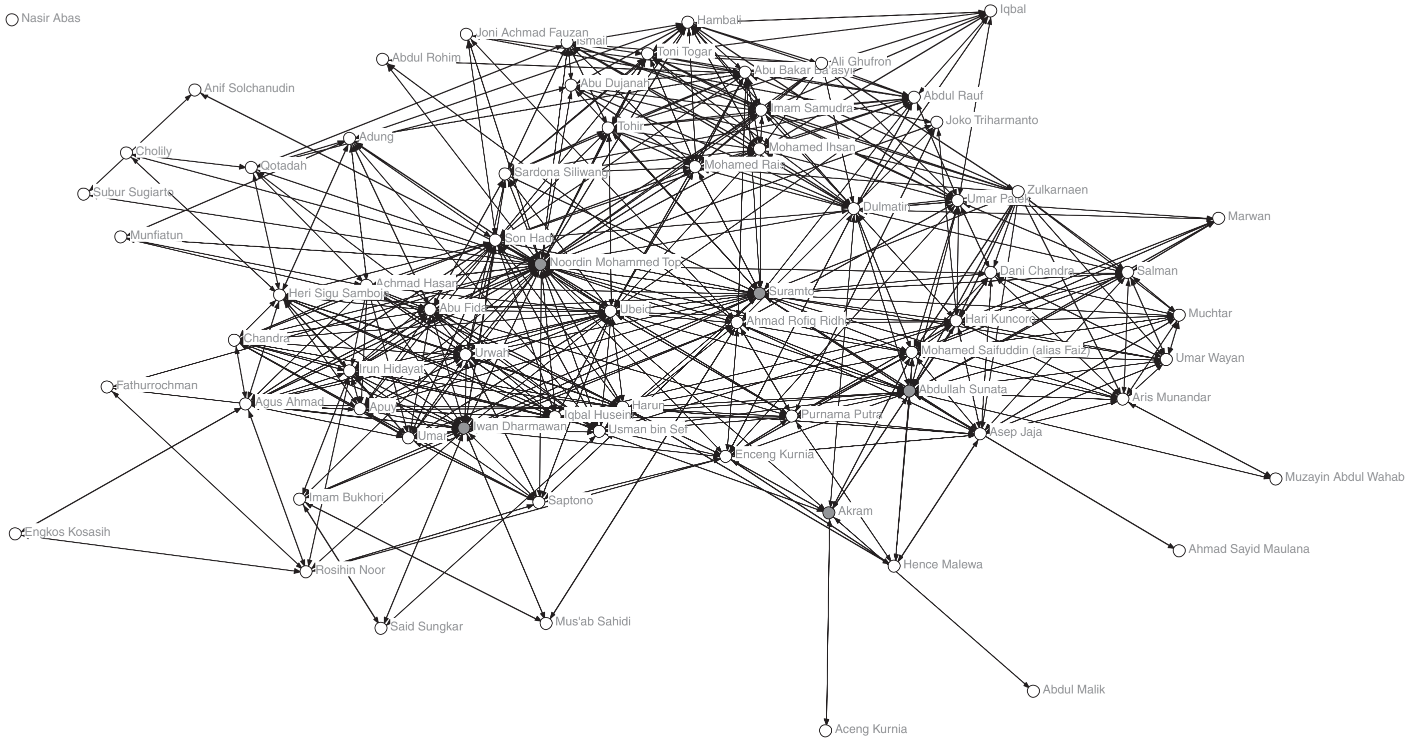


Figure 8.20. Key Players in the Alive Combined Network

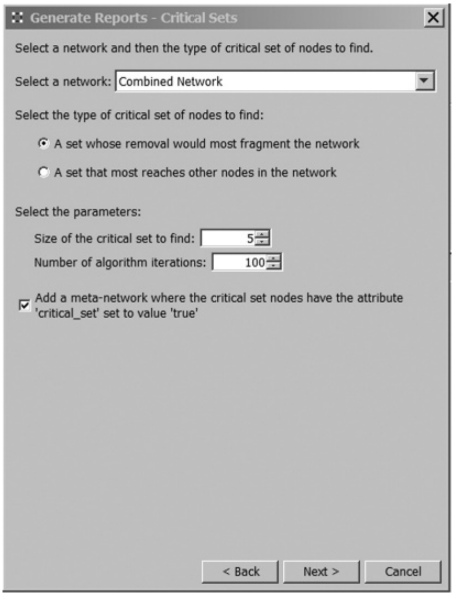


Figure 8.21. ORA Critical Set Dialog Box

of Rosihin Noor and Suramto. There is far less agreement between the two reach algorithms. They only identify two of the same individuals (Noordin Top and Abdullah Sunata) as ideal candidates for the diffusion of misinformation, surveillance, or attempting to turn. This, of course, is not entirely unexpected because they do operate under slightly different assumptions. Thus, analysts would probably want to compare the results of the two algorithms, consider whether path distance should be treated as an important factor, and consult additional relevant information.

Identifying Key Players (Critical Sets) with ORA

[ORA-Main Screen] Analysis >Generate Reports >Locate Key Entities >Critical Sets

We locate key players in ORA using its *Analysis>Generate Reports >Locate Key Entities>Critical Sets* command. The first dialog box that this command calls up (not shown) asks us to indicate which meta-networks we intend to analyze. Here we will use the alive meta-network (`Alive Noordin Network.xml`). The next dialog box (not shown) asks which networks within each meta-network we intend to analyze. ORA's default settings are typically fine. At the third dialog box (Figure 8.21), however, we will have to change several of ORA's defaults. At the top of the dialog box, we need to indicate which network ("Combined") to use. Then, we have to tell ORA whether we want a set that fragments the network or one that best reaches other actors in the network (recall

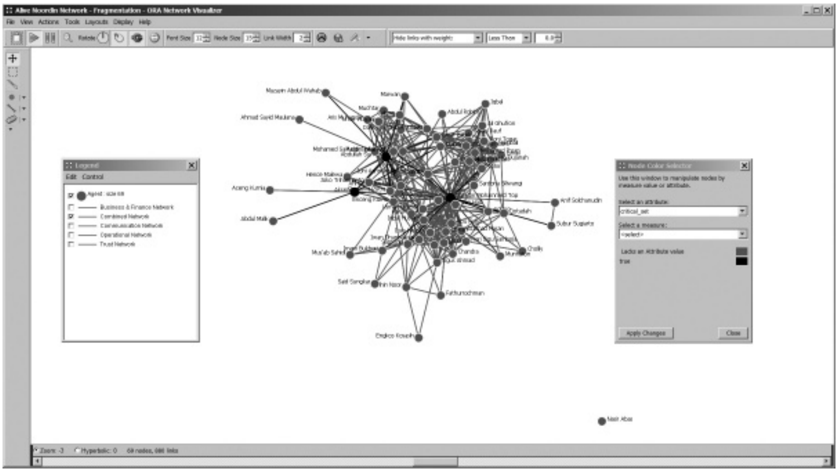


Figure 8.22. ORA Visualization of Critical Set in Combined Alive Network

that ORA does not include distance-weighted algorithms). Next, we have to select the size of the critical set. Finally, we have to choose whether we want ORA to generate a new meta-network that includes a critical set attribute. This is useful for visualization purposes, so we typically will want to select this option as well. Clicking "Next" takes us to the fourth and final dialog box (not shown) where we indicate the type of output we want and where we want any reports to be stored. Clicking "OK" generates a report that contains similar information to the information obtained using Key Player: in particular, the set of critical nodes and before and after fragmentation or reach scores.

The meta-network that includes a critical set attribute that we asked ORA to create should now appear in the Meta-Network Manager. If you visualize this meta-network and adjust the color of the nodes (*Display>Node Appearance>Node Color>Color Nodes by Attribute or Measure*) to reflect whether they are a part of the critical set or not, you should get a network map similar to Figure 8.22. Although the node labels are a little diificult to see, the results are identical to those obtained using Key Player.

[ORA-Visualizer] Display >Node Appearance >Node Color >Color Nodes by Attribute or Measure

8.5 Affiliations and Brokerage

Group affiliation is often an important factor in brokerage processes. For example, in brokering deals in Congress, U.S. senators not only take into account their own interests and desires but also the political party of which they are a part. Although they might want to support a particular

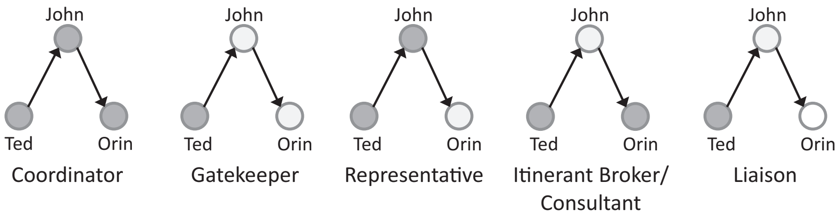


Figure 8.23. Brokerage Roles of "John"

legislative bill, their party membership may constrain what they are able to do and say. Roger Gould and Roberto Gonzalez (Fernandez and Gould 1994; Gould and Fernandez 1989) have attempted to capture this dynamic by identifying five different types of brokerage roles that actors can play based on their group affiliation: (1) coordinator, (2) gatekeeper, (3) representative, (4) itinerant broker/consultant, and (5) liaison (see Figure 8.23 where node color indicates group affiliation).

- *Coordinator* – provides mediation between members of one group where the mediator is also a member of the group
- *Gatekeeper* – provides mediation between two groups where mediator regulates the flow of information or goods to his or her group
- *Representative* – provides mediation between two groups where mediator regulates the flow of information or goods from his or her group
- *Itinerant Broker/Consultant* – provides mediation between members of one group where the mediator is not a member of the group
- *Liaison* – provides mediation between two groups where mediator does not belong to either group

Gould and Fernandez originally conceived of these five types of brokerage roles in terms of directed networks, but tie direction only distinguishes between representative and the gatekeeper brokerage roles. Thus, if we apply their brokerage roles algorithm to undirected networks, each actor identified as a representative will also be a gatekeeper and vice versa. In the following examples, we will return to the alive and free combined network that we analyzed in terms of factions and Newman groups in Chapter 6. If you recall, those two sets of algorithms offered slightly different answers as to which subgroups certain actors belonged, suggesting that they may be in a position of brokerage between such groups. Also, at this writing, only UCINET and Pajek include the Gould and Fernandez algorithm, so we limit our analysis to those two programs.

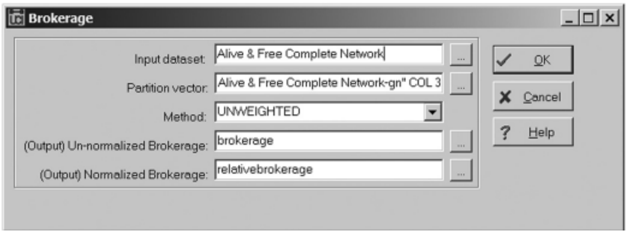


Figure 8.24. UCINET Brokerage Role Dialog Box

Affiliations and Brokerage in UCINET and NetDraw

[UCINET] Network >Ego Networks >G&F Brokerage roles

In UCINET we identify brokerage roles using its *Network>Ego Networks>G&F Brokerage roles* command, which calls up the brokerage dialog box (Figure 8.24).

As you can see, because this routine needs to know the groups to which actors belong, not only do we need to indicate the network we intend to analyze (`Alive & Free Combined Network.##h`) but also a partition file that indicates group membership. This partition can be based on predefined groups (e.g., Republican, Democrat) or on groups detected through one or more of the clustering algorithms we examined in previous chapters. Here, I have chosen the three-group (column 3) Girvan-Newman partition that we generated using UCINET in Chapter 6 (`Alive & Free Combined Network-gn.##h`). Keeping UCINET's defaults and clicking "OK" generates results that are displayed in an output log and saved in two attribute files (`brokerage.##h and relativebrokerage.##h`). UCINET's output log (a portion of which is presented in Figure 8.25) is extensive. It first lists raw brokerage role scores (i.e., the count of how often each actor "plays" a particular role within the network – note that the Gatekeeper and Representative scores are identical). Next, comes a series of block matrices for each actor, wherein each row/column refers to the various groups in the network, and the numbers in the cells indicate how often each actor plays a brokerage role either between two groups or within a single group. Finally, it provides normalized brokerage scores (i.e., raw brokerage role counts are divided by expected counts – given a random network – based on network size).

Table 8.3 summarizes the results of the Gould and Fernandez brokerage analysis. The Gatekeeper and Representative scores have been combined (not summed) into a single column because (as noted previously) with undirected networks, the scores are identical. The "total" score has been adjusted accordingly as well. Not surprisingly, Noordin is found in the most brokerage roles; in fact, he is the only individual to be located in all four of the types of brokerage roles examined here. Next in terms of total count is Suramto, who has turned up in previous analyses as a potential broker. However, he only functions as a coordinator or gatekeeper but

```
ucinetlog7.txt - Notepad
File  Edit  Format  View  Help
GOULD & FERNANDEZ BROKERAGE MEASURES
--------------------------------------------------------------------------------

Input dataset:                      Alive & Free Complete Network (E:\Sociology\Social Net
Partition vector:                   "Alive & Free Complete Network-gn" COL 3
Method:                             UNWEIGHTED
Normalized Brokerage:               relativebrokerage (E:\Sociology\Social Network Analysi
Unnormalized Brokerage:             brokerage (E:\Sociology\Social Network Analysis\Book\D
Warning: Attribute vector has been recoded.

Here is a translation table:

  Old Code     New Code   Frequency
  ========     ========   =========
      1    =>      1          1
      2    =>      2          5
      3    =>      3         10
      4    =>      4          1
      5    =>      5          4
      6    =>      6          1
      7    =>      7          1
      8    =>      8          1

Number of classes: 8

Un-normalized Brokerage Scores

                                  1         2         3         4         5         6
                          Coordinat Gatekeepe Represent Consultan   Liaison     Total
                          -------------------------------------------------------------
 1           Abdul Malik |        0         0         0         0         0         0 |
                         ---------------------------------------------------------------
 2           Abdul Rohim |        0         0         0         0         0         0 |
15 Noordin Mohammed Top  |       12        19        19         2         8        60 |
16               Qotadah |        0         0         0         0         0         0 |
 6               Chandra |        0         0         0         0         0         0 |
21        Subur Sugiarto |        0         0         0         0         0         0 |
                         ---------------------------------------------------------------
 5          Aris Munandar |      10         0         0         0         0        10 |
 7              Dulmatin |        8         8         8         0         0        24 |
11               Muchtar |        0         0         0         0         0         0 |
10          Hari Kuncoro |       14         0         0         0         0        14 |
23            Umar Patek |        0         2         2         0         0         4 |
24            Zulkarnaen |       14         0         0         0         0        14 |
13   Muzayin Abdul Wahab |        0         0         0         0         0         0 |
 3           Abu Dujanah |        2         4         4         0         0        10 |
```

Figure 8.25. UCINET Brokerage Role Output Log

not as a consultant or liaison. Next comes Dulmatin, whom, you may recall from Chapter 6, we identified along with Umar Patek as a possible broker. Interestingly, what this analysis seems to suggest is that while our "intuition" regarding Dulmatin was correct, it was incorrect regarding Patek. Patek may be important to Noordin's (and others') network in some fashion, but at least in terms of the combined alive and free network, he is not.

[NetDraw] File>Open >Uninet data >Network File >Open >Uninet data >Attribute data

Although NetDraw does not include the Gould and Fernandez brokerage role routine, we can still use it to visualize the number of brokerage roles for each actor by varying the size of each node by one (or total) of the brokerage scores. Figure 8.26 does just this where the size of the node indicates the Gatekeeper brokerage score. Moreover, the color of the nodes indicates the Girvan-Newman group to which they were assigned and used to calculate the brokerage scores. This network map was created by first loading the combined alive network into NetDraw, and then reading in both the brokerage scores file (`brokerage.##h`) and the Girvan-Newman partition (`Alive & Free Combined Network-gn.##h`).

Properties >Nodes> Symbols >Color >Attribute-based

Then, the size of the nodes were adjusted using the *Properties>Nodes> Symbols>Size>Attribute-based* command and the color of the nodes was adjusted using the *Properties>Nodes>Symbols>Color>Attribute-based* command. The graph aptly illustrates the results of Table 8.3. Indeed, it amplifies the position of brokerage that Noordin holds between various

Table 8.3. *Gould and Fernandez brokerage scores for alive combined network*

	Coordinator	Gatekeeper/ representative	Itinerant broker consultant	Liaison	Total
Abdul Malik	0	0	0	0	0
Abdul Rohim	0	0	0	0	0
Abu Dujanah	2	4	0	0	6
Aceng Kurnia	0	0	0	0	0
Aris Munandar	10	0	0	0	10
Chandra	0	0	0	0	0
Dulmatin	8	8	0	0	16
Enceng Kurnia	0	5	0	2	7
Engkos Kosasih	0	0	0	0	0
Hari Kuncoro	14	0	0	0	14
Muchtar	0	0	0	0	0
Mus'ab Sahidi	0	0	0	0	0
Muzayin Wahab	0	0	0	0	0
Nasir Abas	0	0	0	0	0
Noordin Top	12	19	2	8	41
Qotadah	0	0	0	0	0
Rosihin Noor	4	0	0	0	4
Said Sungkar	0	0	0	0	0
Salman	14	0	0	0	14
Saptono	0	3	0	0	3
Subur Sugiarto	0	0	0	0	0
Suramto	8	11	0	0	19
Umar Patek	0	2	0	0	2
Zulkarnaen	14	0	0	0	14

subgroups in the network (recall that the gatekeeper brokerage role captures an actor's mediation potential between groups). It also highlights that Suramto, Dulmatin, and possibly Enceng Kurnia may be in positions to mediate between groups as well. Of course, this only illustrates the relative potential of actors in terms of gatekeeping; similar network maps can be generated in terms of the other brokerage roles as well.

Affiliations and Brokerage in Pajek

In Pajek the *Operations>Brokerage Roles* command identifies brokerage roles. After opening the alive and free project file (`Alive and Free Noordin Network.paj`), make sure that the network of interest (e.g., alive and free combined network) is displayed in the first Network drop-down menu and the Girvan-Newman partition is displayed in the first Partition drop-down menu, and then issue the command. This will generate five new partitions, one for each type of brokerage role, all of which

[Pajek] Operations >Brokerage Roles

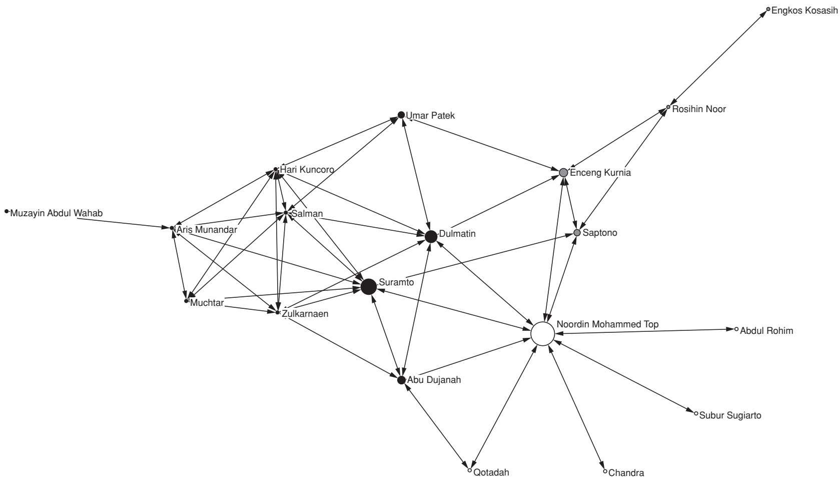


Figure 8.26. NetDraw Drawing of Gatekeeper Roles in Combined Alive Network

are added to the Partition drop-down menu. The class number assigned to an actor in each partition specifies the number of times that actor is found in the corresponding brokerage role.

Info>Partition

We can call up a frequency table for each partition using the *Info >Partition* command, and we can check the individual scores of each actor using Pajek's *File>Partition>Edit* command. To visualize the network where node size reflects the number of brokerage roles, we need to first convert the brokerage partitions into vectors using the *Partition>Make Vector* command. Then we can visualize the network similarly to how we did with NetDraw (*Draw>Draw-Partition-Vector*). Specifically, if we highlight the combined alive and free network in the first Network drop-down menu, the Girvan-Newman partition in the first Partition drop-down menu, and a vector generated from the Gatekeeper partition, we will produce a network map similar to Figure 8.27, which, while not identical to Figure 8.26, tells a very similar story.

File>Partition >Edit

Draw>Draw-Partition-Vector

8.6 Bridges and Network Flow

Perhaps one of the most parsimonious means for detecting bridges within a network is by focusing on the potential flow of resources through a network. Recall from Chapter 6 that one way to identify Newman groups (Girvan-Newman – the approach that UCINET uses) is to first estimate betweenness centrality for each edge/tie (not actor) in the network, remove the tie with the highest betweenness score, recalculate edge

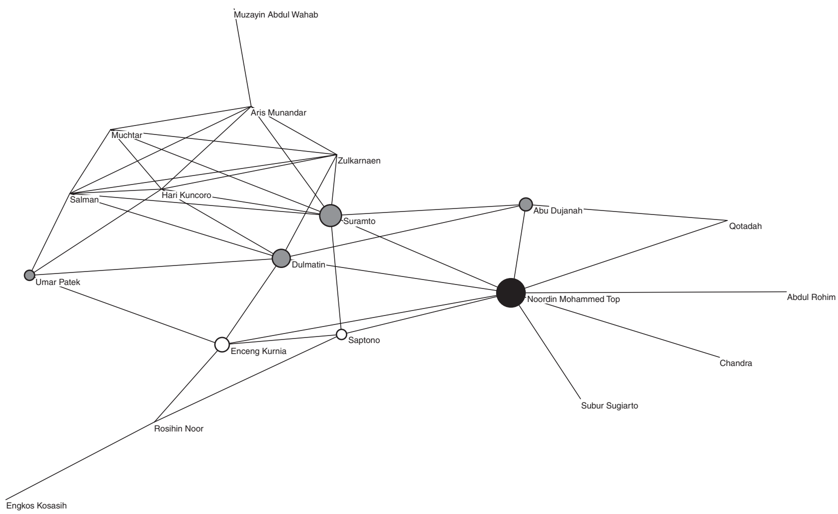


Figure 8.27. Pajek Drawing of Gatekeeper Roles in Combined Alive Network

betweenness centrality, and then iteratively repeat the process until no edges/ties remain. Because edges of high betweenness are assumed to be vital for connecting different parts of the network, it is an efficient method for breaking the network down into large subgroups and then smaller ones. An implicit assumption of this approach is that edges that score high in terms of edge betweenness are likely to span gaps in the social structure, what Ron Burt calls structural holes. In other words, edges that score high in terms of edge betweenness are more likely to be bridges than those that do not.

Edge Betweenness in UCINET and NetDraw

Currently, only UCINET estimates edge betweenness, so we will focus our efforts in UCINET and NetDraw, using the combined alive and free network as an example. Edge betweenness is calculated in UCINET using its *Network>Centrality and Power>Freeman Betweenness>Edge (line) Betweenness* command. Generally, we will want to accept UCINET's defaults and click "OK." The command generates a new network where the cell values are edge betweenness scores. If we examine the network matrix, we can determine which edge has the highest betweenness score.

[UCINET] Network >Centrality and Power >Freeman Betweenness >Edge (line) Betweenness

Unfortunately, while the inspection of a network matrix of edge betweenness scores is feasible with small networks, it becomes increasingly difficult with large networks. An easier way to examine the network is to visualize it in NetDraw. Open your newly generated network

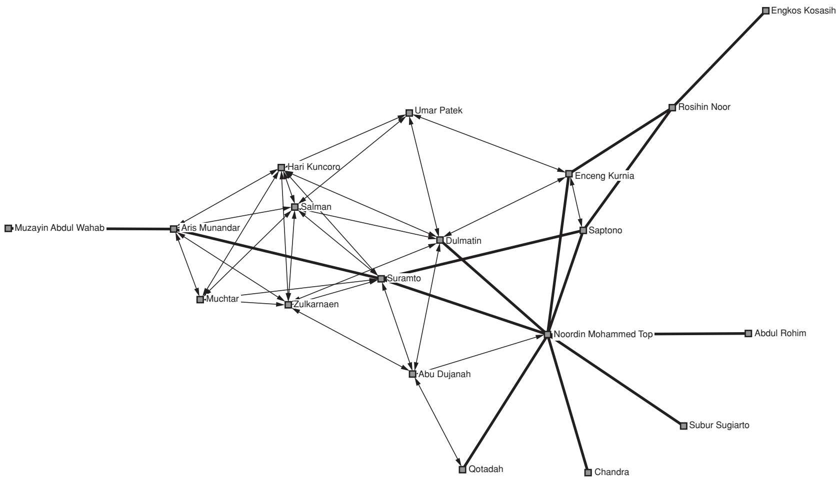


Figure 8.28. Combined Alive Network, Tie-Width Equals Edge Betweenness (NetDraw)

[NetDraw] Properties >Lines >Size >Tie Strength

(probably called "EdgeBetweeness") in NetDraw and vary the size of the ties based on tie strength (*Properties>Lines>Size>Tie Strength*). To see the edges clearly you may need to minimize the size of the labels (or hide them altogether), but you should end up with a network map that looks similar to Figure 8.28. As you can see, some of the ties score higher in terms of edge betweenness than do others. We can probably ignore some because they are ties to individuals with only one tie to the network (known as pendants), but others (e.g., Noordin-Suramto, Suramto-Aris Munandar, Suramto-Saptono) could be of possible interest.

One of the attractions of detecting bridges is that dissolving a tie may, in some cases, be easier to do than removing a particular actor from a network. Why? One reason is that there are always two actors involved in every bridge, and one may be easier to access than the other (e.g., one may be less central than the other or may carry out a role that has higher visibility). Another is that a tie can be dissolved through a misinformation campaign that seeks to create distrust between those on either side of the bridge or through an information operations campaign that attempts to sever the ability of certain pairs of actors to communicate.

8.7 Summary and Conclusion

In this chapter we have examined five approaches for identifying brokers and/or bridges in a network. We began by looking at Burt's (1992a, b) notion of structural holes, which calculates the level of constraint each actor in a network faces. As we saw, it builds on Mark Granovetter's (1973, 1974) notion of weak ties but takes the position that when it comes to identifying brokerage potential, it is not the type of tie that is important but rather the gaps in the social structure. Next we examined how we can use a technique known as bi-component analysis to identify the actors (cutpoints) and bridges (bi-components of size two) within a network whose removal will disconnect it. We then explored Borgatti's (2006) key player algorithms, which identify optimal sets of actors that can be targeted for either fragmenting the network or for diffusing resources through a network. His approach represents an improvement over bi-component analysis because in well-connected networks, cutpoints often do not exist. After this we turned to the Gould and Fernandez algorithms (Fernandez and Gould 1994; Gould and Fernandez 1989) that assume that brokerage is a function of the different groups with which actors are affiliated; thus, not only does this approach require network data, it also requires attribute data indicating the specific groups to which actors belong. Finally, we considered an approach that draws on the notion of betweenness centrality in order to identify ties in a network that are more likely to be functioning as bridges in the network, that is, they are more likely to be the ties through which material and nonmaterial resources flow through a network.

One thing that should be apparent from this chapter's examples is that these algorithms do not always yield the same results. In terms of Noordin's network we identified a number of different individuals (and ties) who may be in a position of brokerage (or a bridge) and therefore warrant closer observation, monitoring, and consideration for the crafting of strategies. However, because the results do vary (and would probably vary even more if we examined all five networks in terms of every possible state of "being" (i.e., alive, alive and free, incarcerated, all). What this highlights is a point that was made in the second chapter: Although social network analysis is a useful tool for the crafting of strategies, a single "magic bullet" algorithm on which we can consistently rely does not exist. Instead, we need to draw on other available information to inform any final decisions that are made.