Architecting Resilient Systems with Design Structure Matrices and Network Topology Analysis

Robert Hill
Thomas Mazzuchi, D.Sc.
Shahram Sarkani, Ph.D., PE
George Washington University

NDIA 2018 Systems Engineering Conference

Why resilience?

"There was little compartmentalization, and each cell mixed freely with the others…. Indeed, the operation succeeded because they did not follow their own rules. Because most of the planners, including the field coordinator and principal executors, were from the same clique and informally benefitted from the free flow of information, they were able to overcome the myriad obstacles they encountered… The success of these operations may be due to their violations of their own operational guidelines."

-Marc Sageman
Understanding Terror Networks

Source:
Sageman, Marc. *Understanding Terror Networks*. University of Pennsylvania Press, 2004, p.167.

Agenda

Resilience as (non-functional) system life cycle property

Agenda

Resilience as (non-functional) system life cycle property

Life cycle properties as functions of architecture

Agenda

Resilience as (non-functional) system life cycle property

Life cycle properties as functions of architecture

Architectures as networks

Agenda

Resilience as (non-functional) system life cycle property

Life cycle properties as functions of architecture

Architectures as networks

Network properties as functions of network topology

Agenda

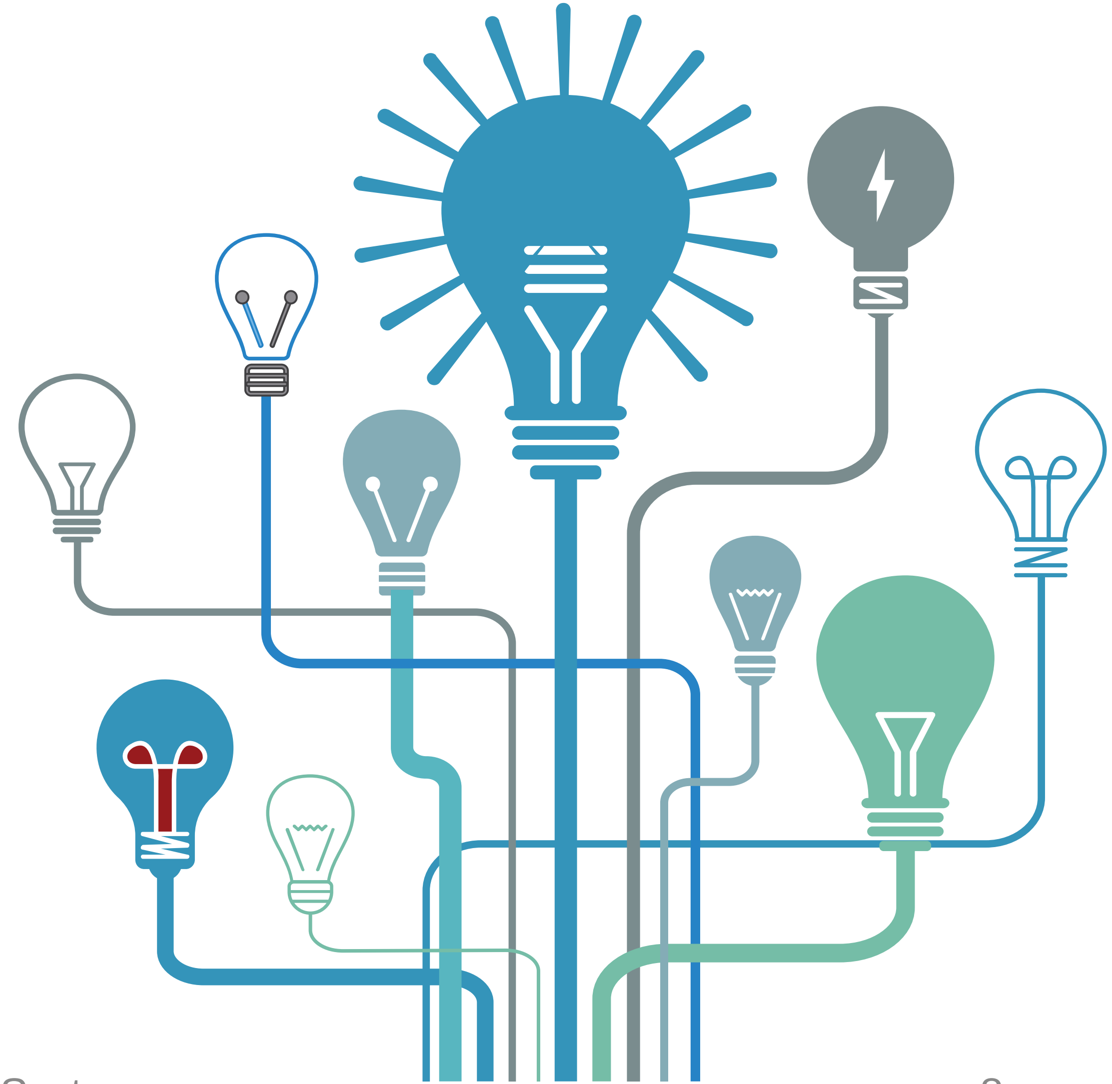
- Resilience as (non-functional) system life cycle property
- Life cycle properties as functions of architecture
- Architectures as networks
- Network properties as functions of network topology

Resilience as function of network topology

Life cycle properties as functions of architecture

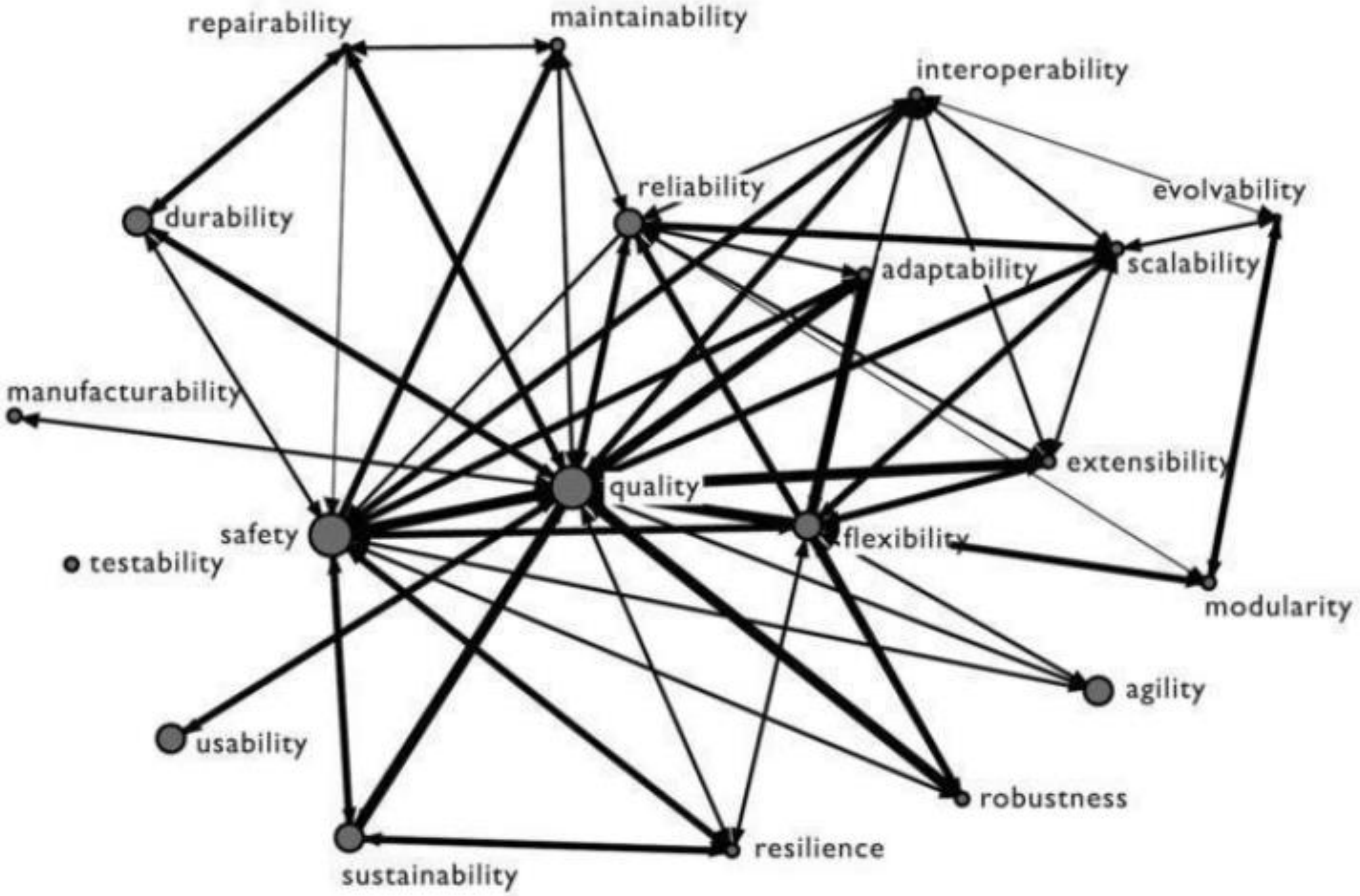
"We now understand better that the life cycle properties of systems (e.g. the ability of a system to be resilient to random or targeted attacks, or its ability to evolve) are largely determined by their underlying architecture."

-Olivier de Weck
Editor-in-chief of *Systems Engineering* (2013-18)
in May 2018 20th anniversary special issue



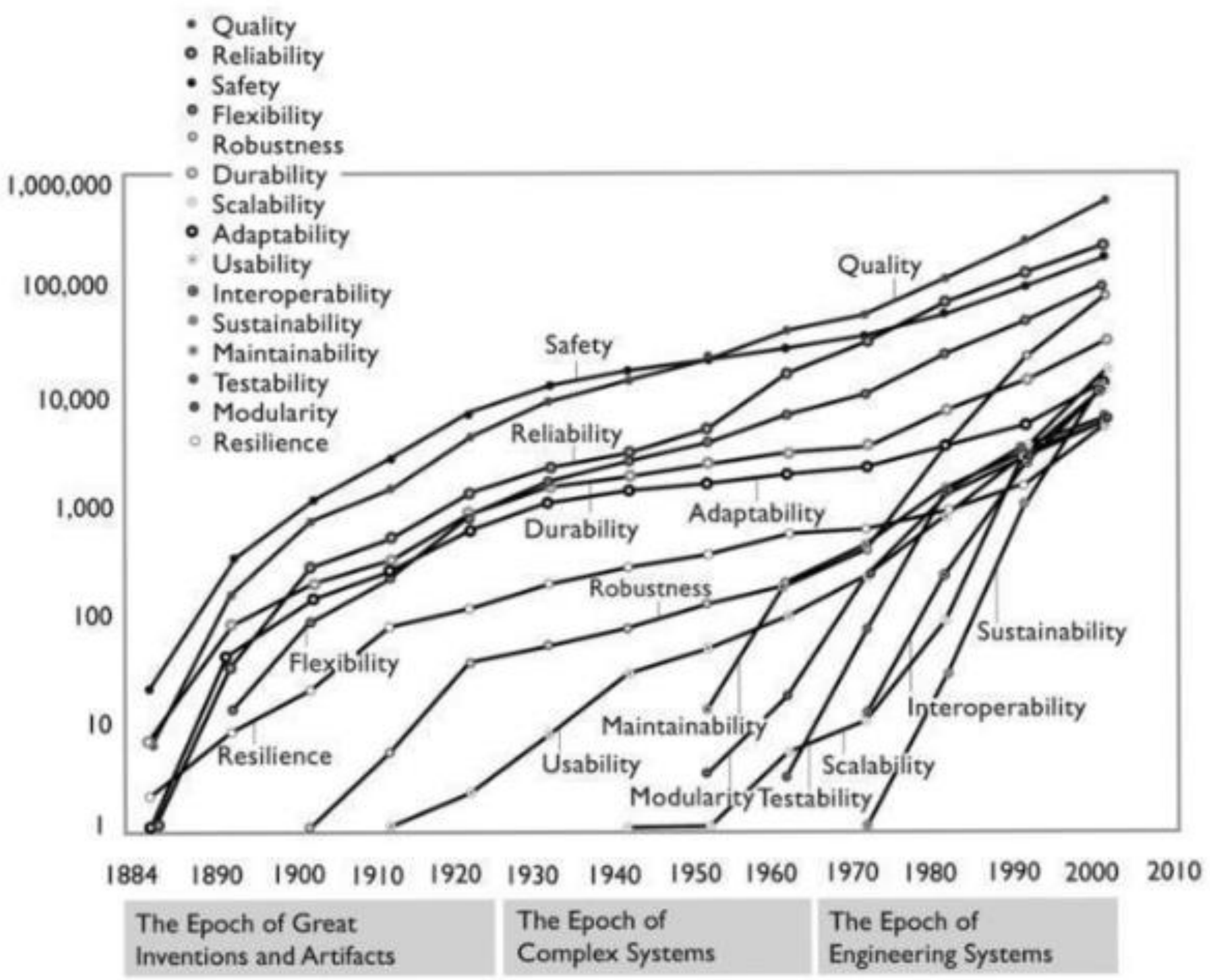
Source:
De Weck OL. Systems engineering 20th anniversary special issue. Systems Engineering. 2018;21:143–147. https://doi.org/10.1002/sys.21443

Resilience as system life cycle property



Source:
De Weck, Olivier L, et al. *Engineering Systems: Meeting Human Needs in a Complex Technological World*. MIT Press, 2016.

Resilience as system life cycle property



Source:
De Weck, Olivier L, et al. *Engineering Systems: Meeting Human Needs in a Complex Technological World*. MIT Press, 2016.

Resilience as system life cycle property

"Systems are no longer just conceived, designed, implemented, and operated in a linear fashion to satisfy stakeholder needs. They are ever-changing, coalescing into systems-of-systems driven by dynamic technological, economic and political forces, and they require us to constantly reassess, upgrade, and evolve them over time. That is why designing systems for specific desired life cycle properties such as *resilience*, *sustainability*, and *evolvability* is more important today than ever before."

-Olivier de Weck

Source:
De Weck OL. Systems engineering 20th anniversary special issue. Systems Engineering. 2018;21:143–147. https://doi.org/10.1002/sys.21443

Life cycle properties as functions of architecture

"Today's systems exist in an extensive network of interdependencies as a result of opportunities afforded by new technology and by increasing pressures to become faster, better and cheaper for various stakeholders. But the effects of operating in interdependent networks has also created unanticipated side effects and sudden dramatic failures. These unintended consequences have led many different people from different areas of inquiry to note that some systems appear to be more resilient than others. This idea that systems have a property called 'resilience' has emerged and grown extremely popular in the last decade…"

-David D. Woods, co-author, *Resilience Engineering: concepts and precepts*

Source:
Woods, David D. "Four Concepts for Resilience and the Implications for the Future of Resilience Engineering." *Reliability Engineering & System Safety*, vol. 141, 2015, pp. 5–9., doi:10.1016/j.ress.2015.03.018.

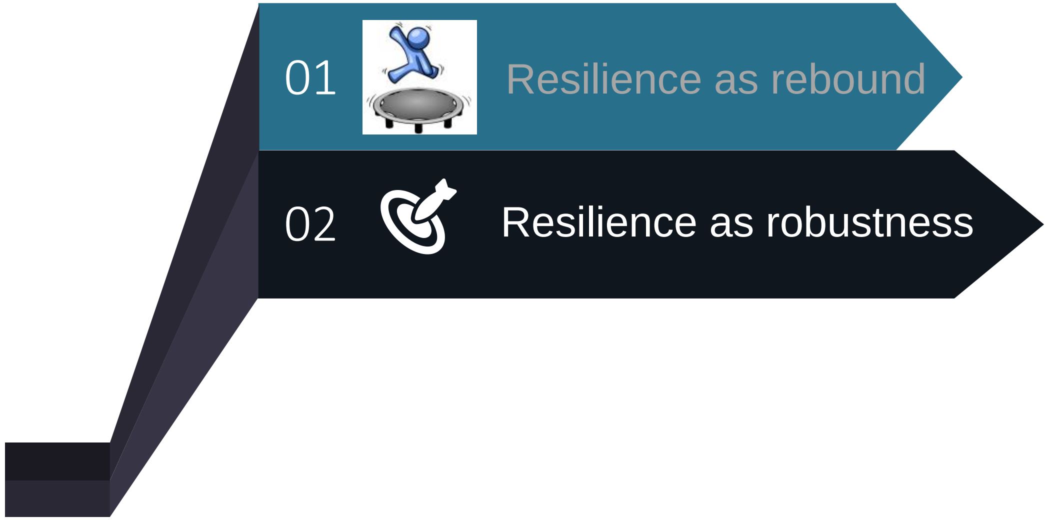
Life cycle properties as functions of architecture

01  Resilience as rebound

"This use of the label resilience as [1] – rebound – is common, but pursuing what produces better rebound merely serves to restate the question…."

Source:
Woods, David D. "Four Concepts for Resilience and the Implications for the Future of Resilience Engineering." *Reliability Engineering & System Safety*, vol. 141, 2015, pp. 5–9., doi:10.1016/j.ress.2015.03.018.

Life cycle properties as functions of architecture



"Resilience [2] – increased ability to absorb perturbations – confounds the labels robustness and resilience… this confound continues to add noise to work on resilience…"

Source:
Woods, David D. "Four Concepts for Resilience and the Implications for the Future of Resilience Engineering." *Reliability Engineering & System Safety*, vol. 141, 2015, pp. 5–9., doi:10.1016/j.ress.2015.03.018.

Life cycle properties as functions of architecture

01  Resilience as rebound

02  Resilience as robustness

03  Resilience as extensibility

"The third concept sees resilience as the opposite of brittleness, or, how to extend adaptive capacity in the face of surprise. Resilience [3] juxtaposes brittleness versus graceful extensibility…"

Source:
Woods, David D. "Four Concepts for Resilience and the Implications for the Future of Resilience Engineering." *Reliability Engineering & System Safety*, vol. 141, 2015, pp. 5–9., doi:10.1016/j.ress.2015.03.018.

Life cycle properties as functions of architecture

"Resilience [4] refers to the ability [to] manage / regulate adaptive capacities of systems that are layered networks, and are also a part of larger layered networks, so as to produce sustained adaptability over longer scales…"

Source:
Woods, David D. "Four Concepts for Resilience and the Implications for the Future of Resilience Engineering." *Reliability Engineering & System Safety*, vol. 141, 2015, pp. 5–9., doi:10.1016/j.ress.2015.03.018.

Architectures as networks

“Technical systems have network structures.
Social, organizational, and technical elements of most sociotechnical systems are interconnected through exchanges of resources (information, energy, and material) and dependencies among various decision parameters in various stages of systems life cycles. Such dependencies are often not uniform and follow structured patterns that can naturally be modeled using complex networks.”

Heydari & Pennock,
in May 2018 20th anniversary special issue of
Systems Engineering

Source
Heydari, Babak, and Michael J. Pennock. “Guiding the Behavior of Sociotechnical Systems: The Role of Agent-Based Modeling.” Systems Engineering, vol. 21, no. 3, 2018, pp. 210–226., doi:10.1002/sys.21435.

Architectures as networks

"Has systems engineering become less waterfall-driven, process-oriented, and heavyweight, and more agile and model-based…?"

"The trends for technology terms [such as] "network AND systems engineering" [and] "graph AND systems engineering"… all suggest this is true."

- Sarah Sheard, INCOSE Fellow, in May 2018 20th anniversary special issue of *Systems Engineering*

	network (K)	graph AND systems engineering
1998	12.663	1.
1999	12.493	3.352
2000	12.643	1.7
2001	12.457	2.21
2002	13.252	3.926
2003	13.710	1.7
2004	14.116	2.861
2005	14.693	2.972
2006	14.724	3.068
2007	15.087	2.0
2008	15.562	5.144
2009	16.129	4.898
2010	15.835	4.097
2011	15.740	4.711
2012	15.383	4.593
2013	15.376	5.091
2014	15.335	5.246
2015	15.804	8.657
2016	16.400	8.700
2017	17.025	7.037

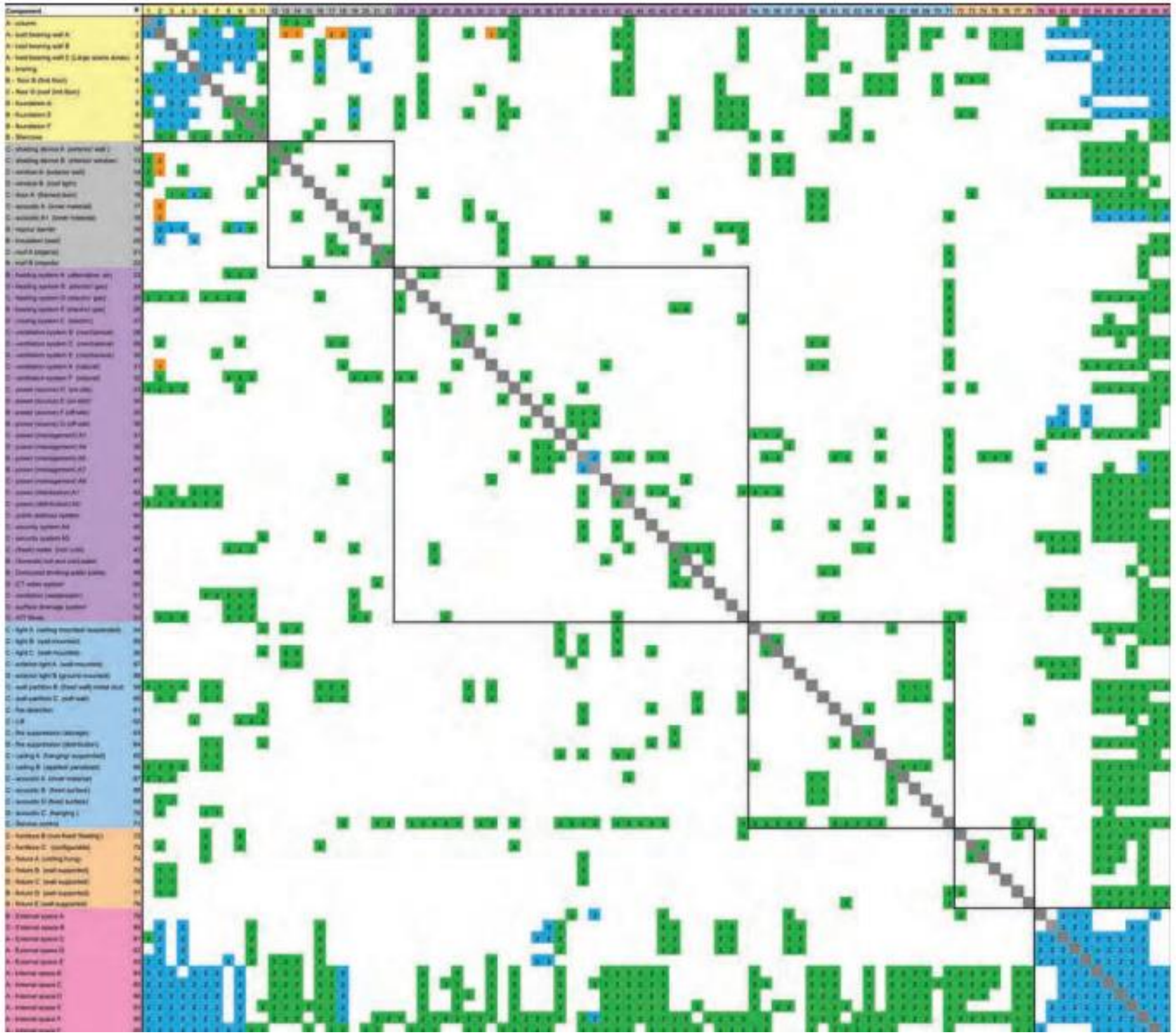
Source
Sheard, Sarah A. "Evolution of Systems Engineering Scholarship from 2000 to 2015, with Particular Emphasis on Software." *Systems Engineering*, vol. 21, no. 3, 2018, pp. 152–171., doi:10.1002/sys.21441.

Architectures as networks: the Design Structure Matrix

"What Is the DSM?

The DSM is a network modeling tool used to represent the elements comprising a system and their interactions, thereby highlighting the system's architecture (or designed structure). DSM is particularly well suited to applications in the development of complex, engineered systems..."

-Eppinger & Browning



Source:
Eppinger, Steven D.; Browning, Tyson R. Design Structure Matrix Methods and Applications. The MIT Press.

Architectures as networks: the Design Structure Matrix

"*DSM* is an *n* x *n* matrix in which rows and columns represent the components and activities within a system. The cell (*i*, *j*) represents the information exchange and dependency patterns associated with the components *i* and *j*. The matrix enables quickly identifying which functions depend on results from which other functions."

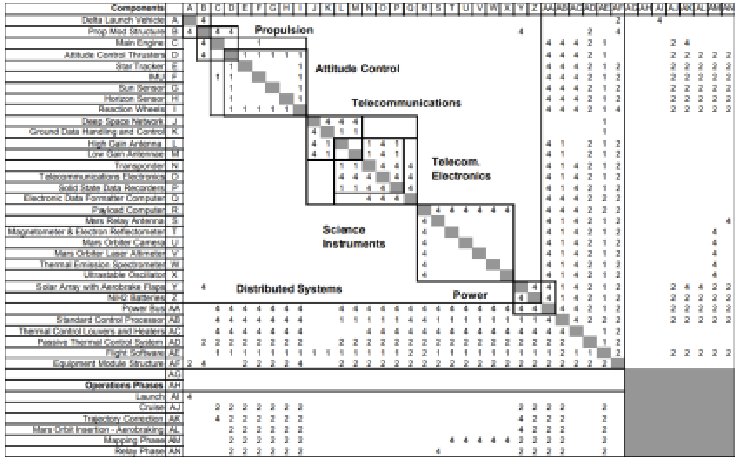
-Madni & Sievers (INCOSE Fellows)
in May 2018 20th anniversary special issue of
Systems Engineering

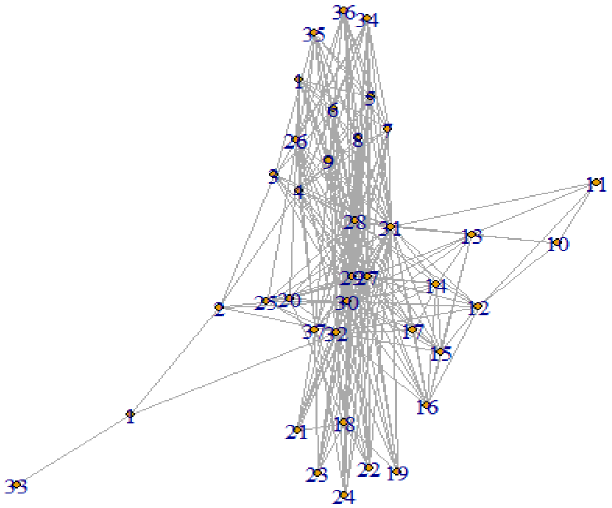
Source:
Madni, Azad M., and Michael Sievers. "Model-Based Systems Engineering: Motivation, Current Status, and Research Opportunities." *Systems Engineering*, vol. 21, no. 3, May 2018, pp. 172–190., doi:10.1002/sys.21438.

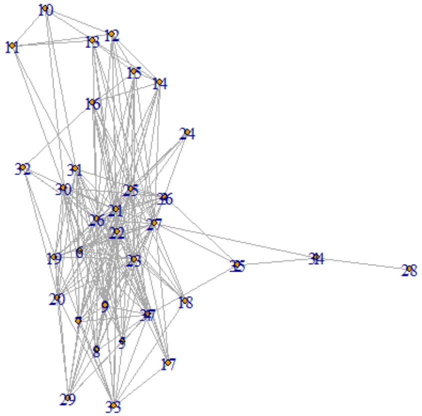
Architectures as networks: the Design Structure Matrix

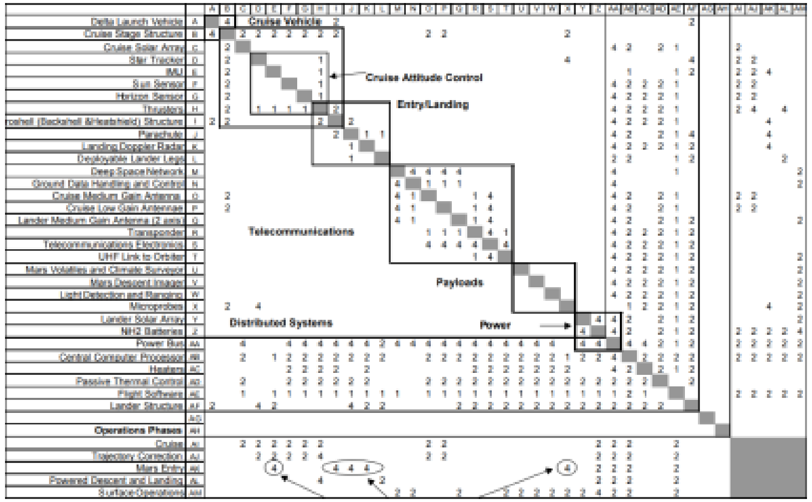
“Compared with other network modelling methods, the primary benefit of DSM is the graphical nature of the matrix display format. The matrix displays a highly compact, easily scalable, and intuitively readable representation of a system architecture...”

-Eppinger & Browning





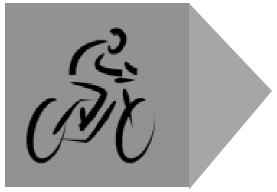




Source:
Eppinger, Steven D.; Browning, Tyson R.. Design Structure Matrix Methods and Applications. The MIT Press. DSM from Tim Brady, MIT Thesis, ‘Utilization of Dependency Structure Matrix Analysis to Assess Implementation of NASA’s Complex Technical Projects’.

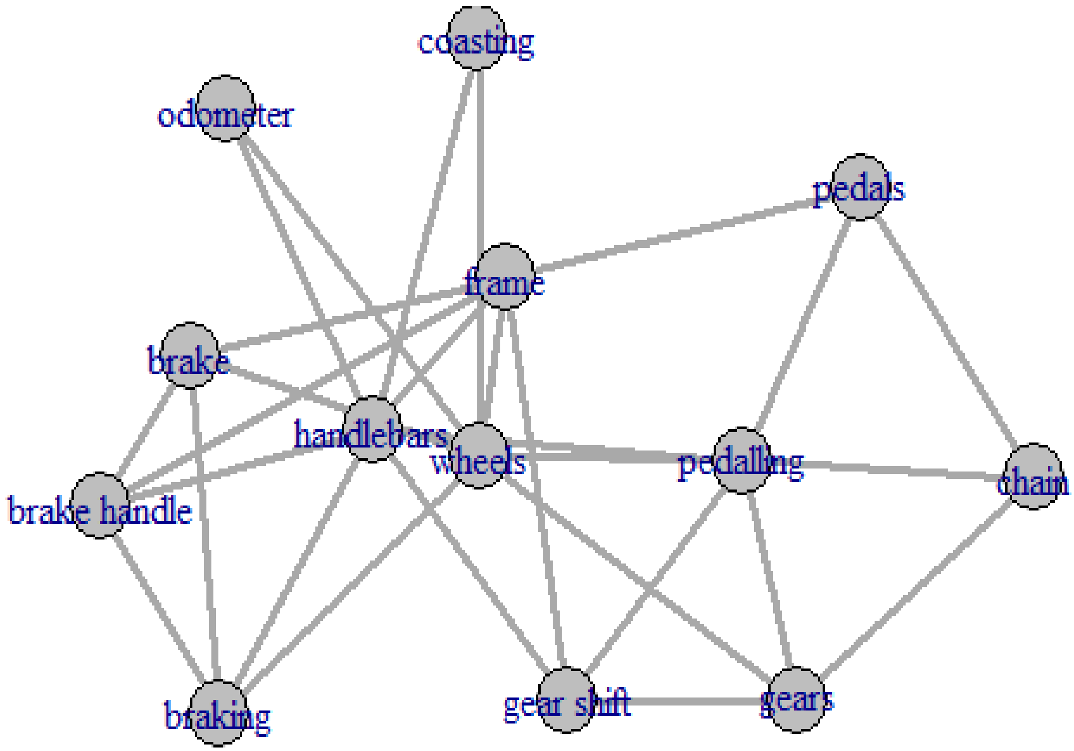
Architectures as networks: the Design Structure Matrix

	Components	Wheels	Gears	Chain	Pedals	Handlebars	Frame	Brake	Brake Handle	Gear Shift	Odometer	Operational Phases	Pedalling	Coasting	Braking
Components		A	B	C	D	E	F	G	H	I	J	K	L	M	N
Wheels	A		4				2	4			2		4	4	4
Gears	B	4		4						4			2		
Chain	C		4		4								4		
Pedals	D			4			2						4		
Handlebars	E						2		2	2	2		2	2	4
Frame	F	2			2	2		2	2	2					
Brake	G	4					2		4						4
Brake Handle	H					2	2	4							4
Gear Shift	I		4			2	2						2		
Odometer	J	2				2									
Operational Phases	K														
Pedalling	L	4	2	4	4	2				2					
Coasting	M	4				2									
Braking	N	4				4		4	4						



"The adjacency matrix [of a network] is simply the binary version of a DSM (placing ones in the cells with marks and zeros elsewhere)."

-Eppinger & Browning



Source:
Eppinger, Steven D.; Browning, Tyson R.. Design Structure Matrix Methods and Applications. The MIT Press. DSM from Tim Brady, MIT Thesis, 'Utilization of Dependency Structure Matrix Analysis to Assess Implementation of NASA's Complex Technical Projects'.

Architectures as networks

The above suggests that while not a universal modeling tool, networks are especially useful as design tools for building specific behaviors, such as resilience, into system architectures, an understudied area of application. The Design Structure Matrix is an especially useful tool in this domain.

Conclusions

- Technical systems have network structure
- Network structure useful for design purposes
- Networks not good models of generic system behavior
- But network statistics useful for modeling specific behaviors
- Networks useful for designing specific behaviors into architecture
- Especially behaviors centered on communication/ exchange of information

Methodology, Findings and Conclusions

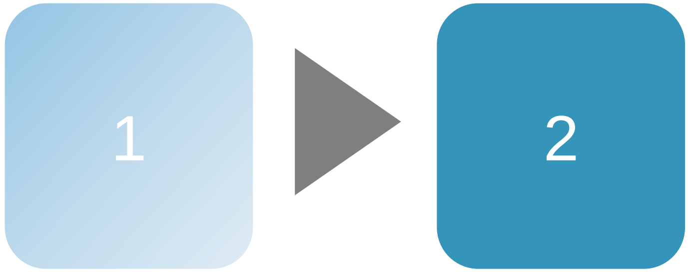
Resilience as function of network topology

Resilience as function of network topology: how to model?

1

Identify networked architectural property as proxy for resilience

Resilience as function of network topology: how to model?



Identify networked architectural property as proxy for resilience

Identify leverage points to achieve architectural property

Resilience as function of network topology: how to model?

1. Identify networked architectural property as proxy for resilience
2. Identify leverage points to achieve architectural property
3. Manipulate leverage points to realize architectural property

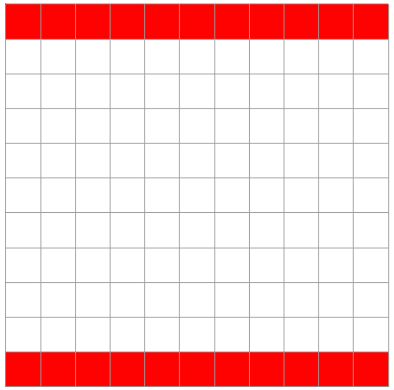
Resilience as function of network topology: how to model?

1. Identify networked architectural property as proxy for resilience
2. Identify leverage points to achieve architectural property
3. Manipulate leverage points to realize architectural property
4. Measure level of realized increase

Resilience as function of network topology: how to model?

1. Identify networked architectural property as proxy for resilience
2. Identify leverage points to achieve architectural property
3. Manipulate leverage points to realize architectural property
4. Measure level of realized increase
5. Simulate effect of increase on network performance

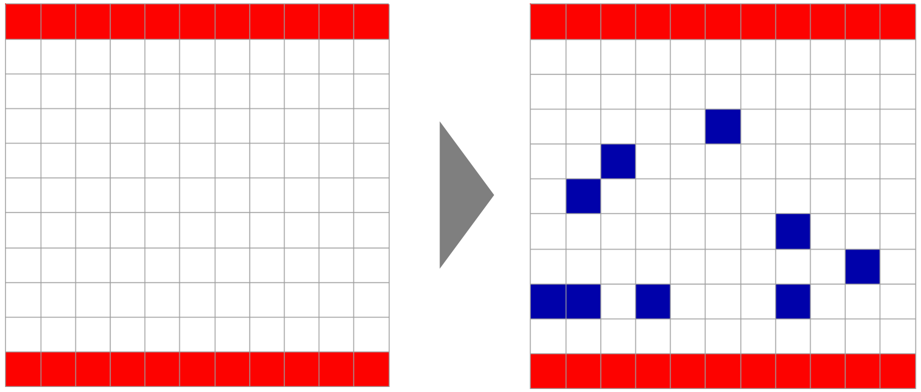
Connectivity phase change on a square lattice



Fluid at top/bottom seeks to percolate through porous medium, e.g. filtration of water through soil or coffee grounds.

Source Code:
"Percolation on a Square Grid" from the Wolfram Demonstrations Project
http://demonstrations.wolfram.com/PercolationOnASquareGrid/
Contributed by: Stephen Wolfram

Connectivity phase change on a square lattice

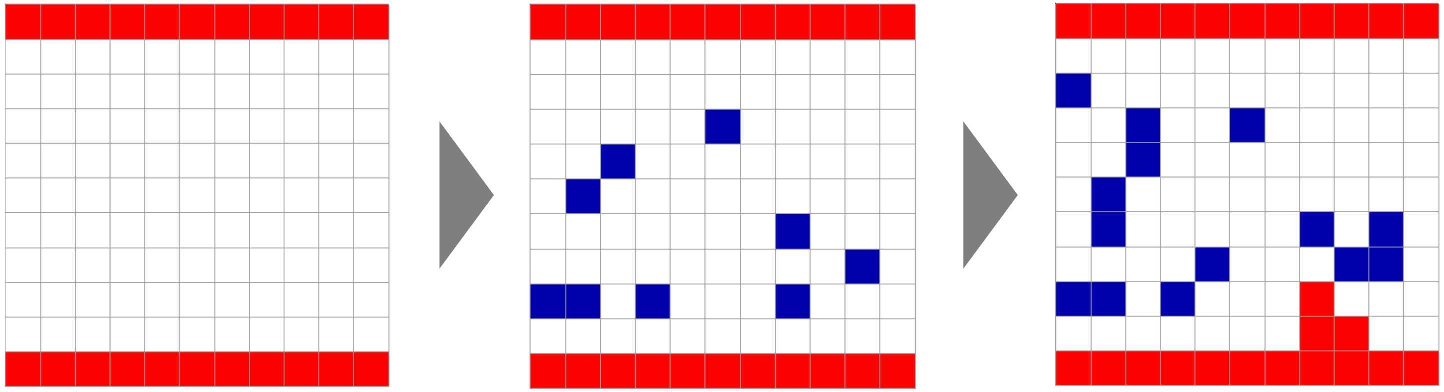


Fluid at top/bottom seeks to percolate through porous medium, e.g. filtration of water through soil or coffee grounds.

At occupation probability p=.1, isolated clusters appear in blue, showing presence of unconnected 'pores'.

Source Code:
"Percolation on a Square Grid" from the Wolfram Demonstrations Project
http://demonstrations.wolfram.com/PercolationOnASquareGrid/
Contributed by: Stephen Wolfram

Connectivity phase change on a square lattice



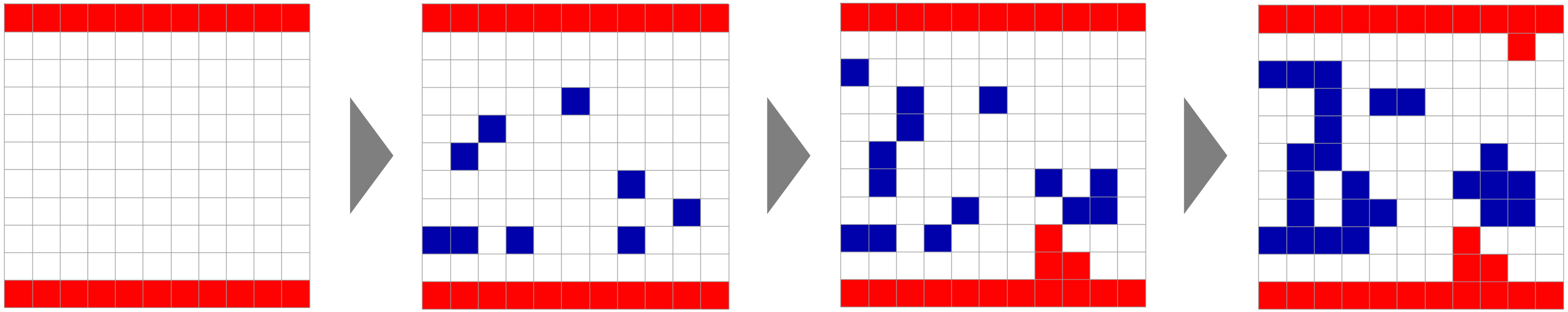
Fluid at top/bottom seeks to percolate through porous medium, e.g. filtration of water through soil or coffee grounds.

At occupation probability p=.1, isolated clusters appear in blue, showing presence of unconnected ‘pores’.

At p=.2, some clusters connect to top/bottom of porous medium, but none span, and no large clusters occur.

Source Code:
"Percolation on a Square Grid" from the Wolfram Demonstrations Project
http://demonstrations.wolfram.com/PercolationOnASquareGrid/
Contributed by: Stephen Wolfram

Connectivity phase change on a square lattice



Fluid at top/bottom seeks to percolate through porous medium, e.g. filtration of water through soil or coffee grounds.

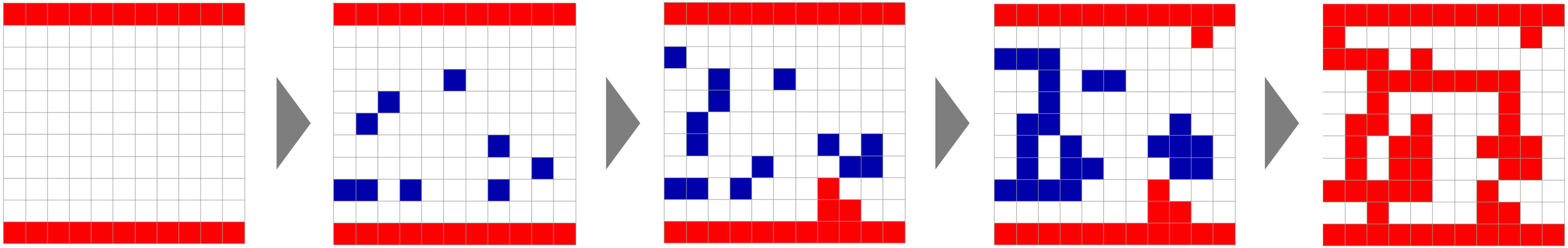
At occupation probability p=.1, isolated clusters appear in blue, showing presence of unconnected 'pores'.

At p=.2, some clusters connect to top/bottom of porous medium, but none span, and no large clusters occur.

At p=.3, multiple large connected clusters emerge, but do not span. Tipping point towards connectivity is imminent.

Source Code:
"Percolation on a Square Grid" from the Wolfram Demonstrations Project
http://demonstrations.wolfram.com/PercolationOnASquareGrid/
Contributed by: Stephen Wolfram

Connectivity phase change on a square lattice



Fluid at top/bottom seeks to percolate through porous medium, e.g. filtration of water through soil or coffee grounds.

At occupation probability p=.1, isolated clusters appear in blue, showing presence of unconnected ‘pores’.

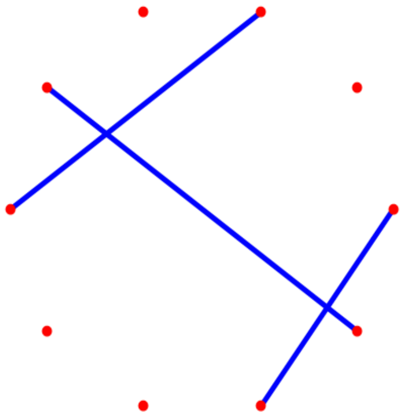
At p=.2, some clusters connect to top/bottom of porous medium, but none span, and no large clusters occur.

At p=.3, multiple large connected clusters emerge, but do not span. Tipping point towards connectivity is imminent.

At p =.4, percolation threshold is passed, and phase transition occurs. Every occupied point on lattice is connected.

Source Code:
"Percolation on a Square Grid" from the Wolfram Demonstrations Project
http://demonstrations.wolfram.com/PercolationOnASquareGrid/
Contributed by: Stephen Wolfram

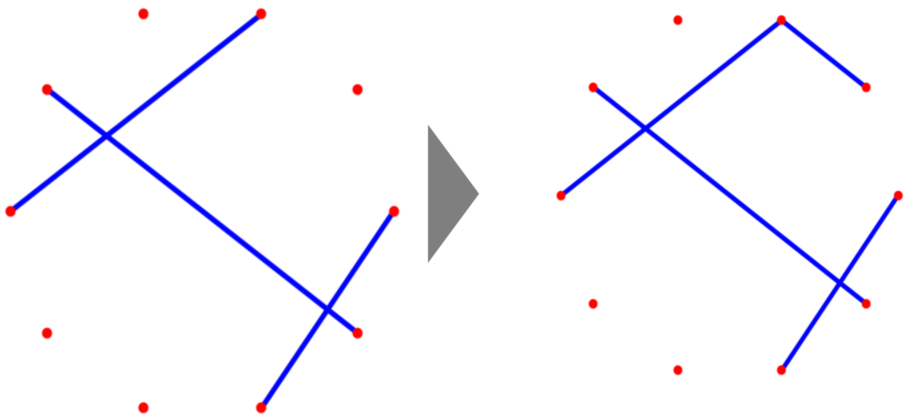
Connectivity phase change on a network



In a random network of 10 nodes, 45 edges [n(n-1)/2)] are possible. Here, there are 3 edges and 3 unconnected clusters.

Source Code:
"Connectivity-Based Phase Transition"
http://demonstrations.wolfram.com/ConnectivityBasedPhaseTransition/ Wolfram Demonstrations Project Published: September 20, 2011
Contributed by: Mark D. Normand

Connectivity phase change on a network

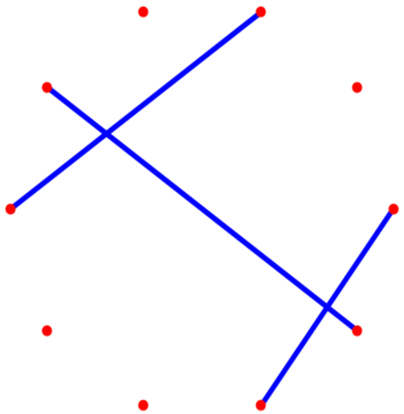


In a random network of 10 nodes, 45 edges [n(n-1)/2)] are possible. Here, there are 3 edges and 3 unconnected clusters.

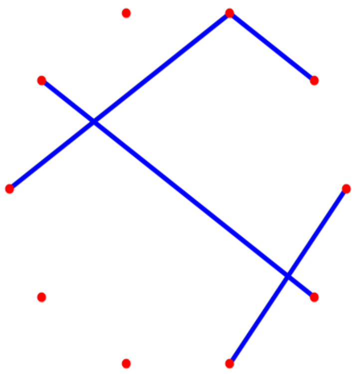
With 4 edges, there are still only 3 unconnected clusters, and largest has only 3 nodes.

Source Code:
"Connectivity-Based Phase Transition"
http://demonstrations.wolfram.com/ConnectivityBasedPhaseTransition/ Wolfram Demonstrations Project Published: September 20, 2011
Contributed by: Mark D. Normand

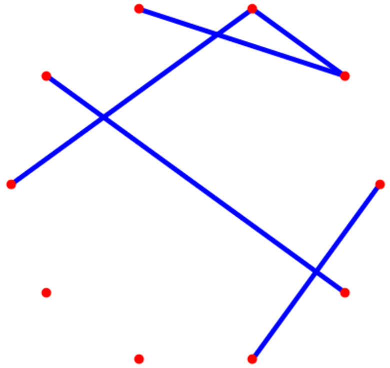
Connectivity phase change on a network



In a random network of 10 nodes, 45 edges [n(n-1)/2)] are possible. Here, there are 3 edges and 3 unconnected clusters.



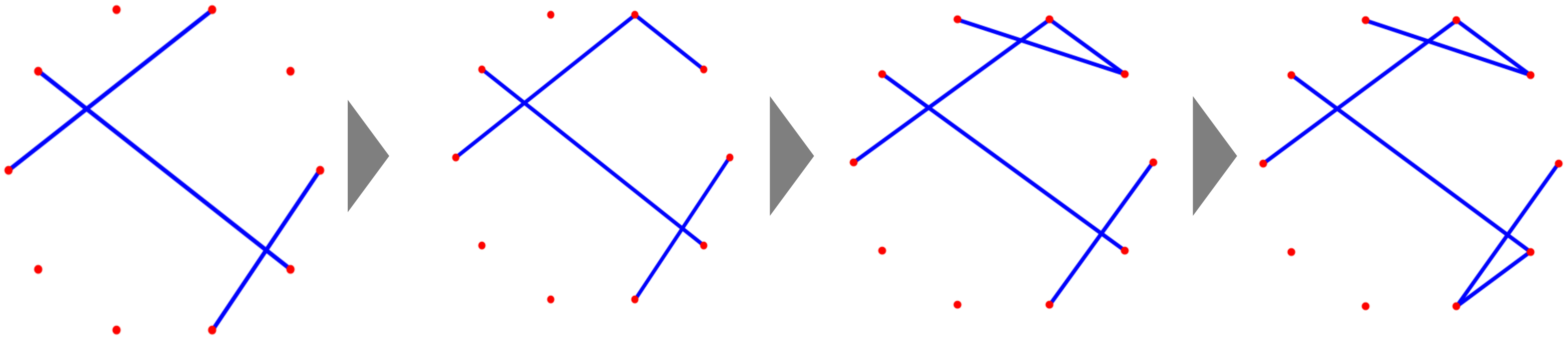
With 4 edges, there are still only 3 unconnected clusters, and largest has only 3 nodes.



With 5 edges, still only 3 clusters, but largest now connects 4 nodes (note: different finite random networks will behave differently).

Source Code:
"Connectivity-Based Phase Transition"
http://demonstrations.wolfram.com/ConnectivityBasedPhaseTransition/ Wolfram Demonstrations Project Published: September 20, 2011
Contributed by: Mark D. Normand

Connectivity phase change on a network



In a random network of 10 nodes, 45 edges [n(n-1)/2)] are possible. Here, there are 3 edges and 3 unconnected clusters.

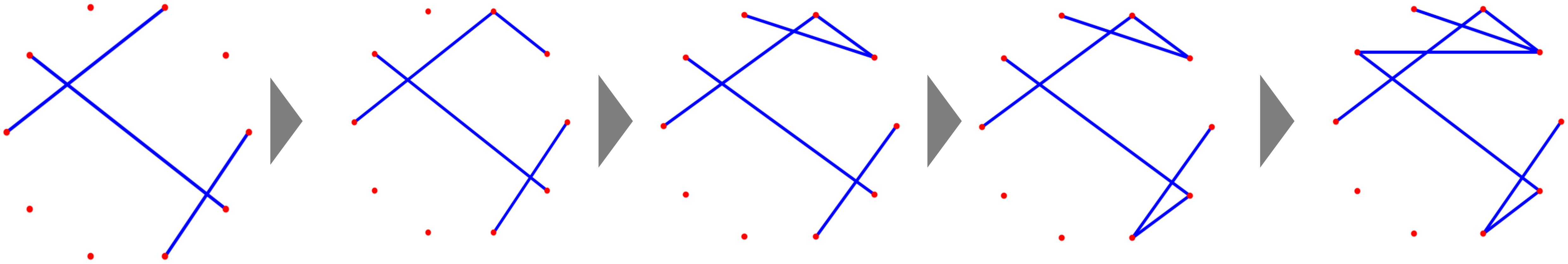
With 4 edges, there are still only 3 unconnected clusters, and largest has only 3 nodes.

With 5 edges, still only 3 clusters, but largest now connects 4 nodes (note: different finite random networks will behave differently).

With 6 edges, two connected clusters emerge, of 4 nodes each. Network is broken into halves, neither communicating.

<u>Source Code:</u>
"Connectivity-Based Phase Transition"
http://demonstrations.wolfram.com/ConnectivityBasedPhaseTransition/ Wolfram Demonstrations Project Published: September 20, 2011
Contributed by: Mark D. Normand

Connectivity phase change on a network



In a random network of 10 nodes, 45 edges [n(n-1)/2)] are possible. Here, there are 3 edges and 3 unconnected clusters.

With 4 edges, there are still only 3 unconnected clusters, and largest has only 3 nodes.

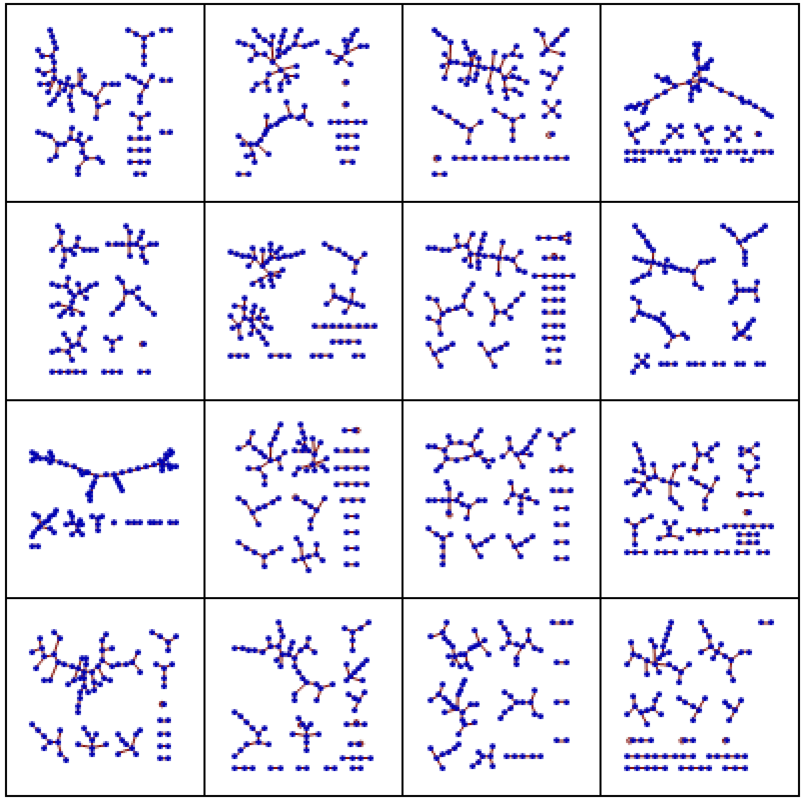
With 5 edges, still only 3 clusters, but largest now connects 4 nodes (note: different finite random networks will behave differently).

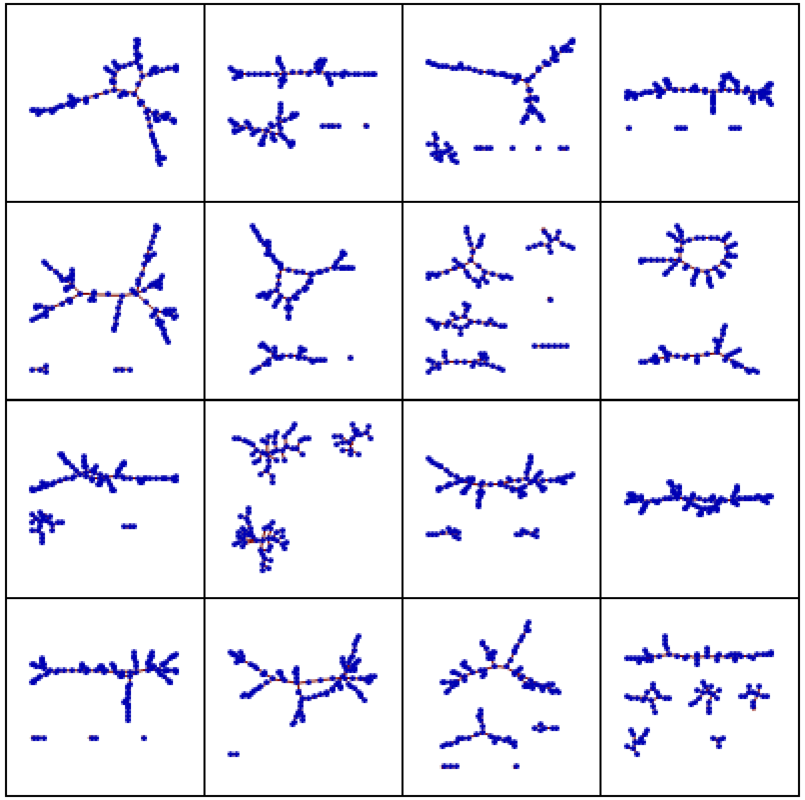
With 6 edges, two connected clusters emerge, of 4 nodes each. Network is broken into halves, neither communicating.

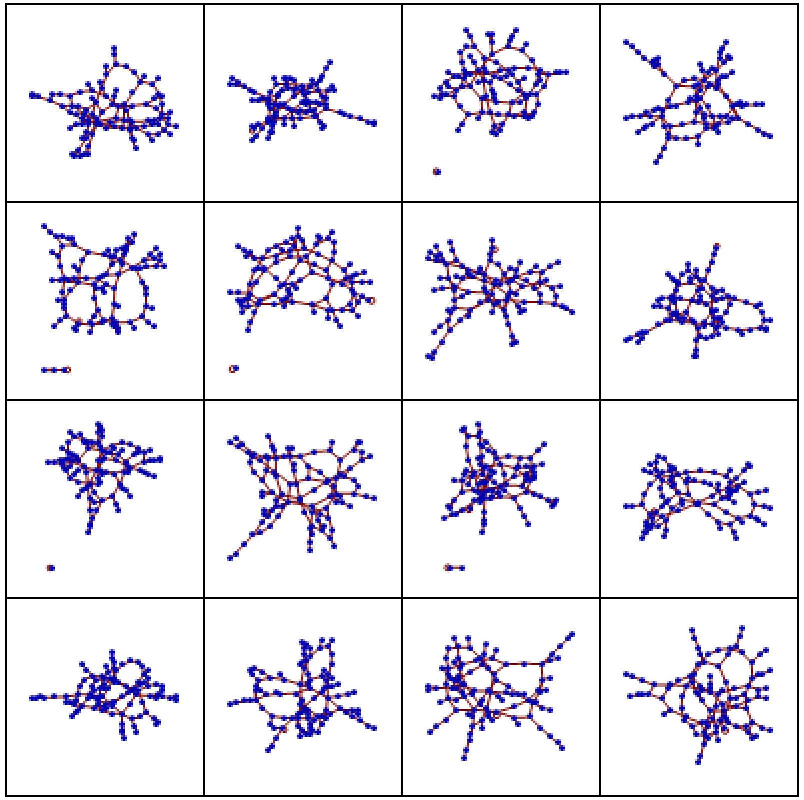
With 7 edges, network ‘tips’ into connectivity, with largest cluster having 8 out of 10 possible nodes (note: network percolation ≠ all terminal connectivity)

Source Code:
"Connectivity-Based Phase Transition"
http://demonstrations.wolfram.com/ConnectivityBasedPhaseTransition/ Wolfram Demonstrations Project Published: September 20, 2011
Contributed by: Mark D. Normand

Connectivity phase changes in random networks







Three arrays of 100-node Erdos-Renyi Poisson random graphs shown, with .8, 1., and 1.2 edges per node, respectively. In the limit of large N, Erdos showed that phase change occurs when average degree of nodes = 1, as shown in middle array. Below that level, graphs are disconnected; above that, connected. At average degree = 1, results are mixed. Note: these regularities do not apply exactly to smaller, finite graphs, as encountered in the real world.

Source Code:
"Samples of Random Graphs" from the Wolfram Demonstrations Project
http://demonstrations.wolfram.com/SamplesOfRandomGraphs/
Contributed by: Stephen Wolfram

Calculating clustering and degree correlation in R

```
> library(igraph)
> vertices=20
> edges=50
> ergraph<-sample_gnm(vertices,edges,directed = FALSE, loops =FALSE)
>
> root_node<-NA
> intermediate_node<-NA
> final_node<-NA
> root_node_degree<-NA
> neighbor_node_degree<-NA
> open_triangle<-NA
> closed_triangle<-NA
> counter=0
> counter2=0
> for (i in 1:vertices) {
+   for (j in (1:length(as_ids(neighbors(ergraph,i))))){
+     counter=counter+1
+     root_node_degree[counter]=degree(ergraph,i)
+     neighbor_node_degree[counter]=degree(ergraph,as_ids(neighbors(ergraph,i))[j])
+     for (k in (1:length(as_ids(neighbors(ergraph,as_ids(neighbors(ergraph,i))[j]))))){
+       counter2=counter2 + 1
+       root_node[counter2]=i
+       intermediate_node[counter2]=as_ids(neighbors(ergraph,i))[j]
+       final_node[counter2]=as_ids(neighbors(ergraph,intermediate_node[counter2]))[k]
+       if (i==final_node[counter2]){
+         open_triangle[counter2]=0
+         closed_triangle[counter2]=0}
+       else if (are.connected(ergraph,i,final_node[counter2])){
+         open_triangle[counter2]=0
+         closed_triangle[counter2]=1}
+       else {
+         open_triangle[counter2]=1
+         closed_triangle[counter2]=0}
+     }}}
```

```
> sum(closed_triangle)/(sum(open_triangle)+sum(closed_triangle))
[1] 0.2288136
> transitivity(ergraph)
[1] 0.2288136
>
> cor(root_node_degree,neighbor_node_degree)
[1] -0.1408083
> assortativity_degree(ergraph)
[1] -0.1408083
```

Estimates of connectivity phase change thresholds in R

```
> library(igraph)
> vertices=1000
> edges=2500
> ergraph<-sample_gnm(vertices,edges,directed = FALSE, loops =FALSE)
> A<-as_adjacency_matrix(ergraph)
> A_eigen<-eigen(A, symmetric = TRUE,only.values=TRUE)[1]
> p_bar<-1/max(A_eigen$values)
> p_bar
[1] 0.1622749
```

For dense matrices, can use inverse of leading eigenvalue of adjacency matrix, as extracted from DSM [due to Bollobas]

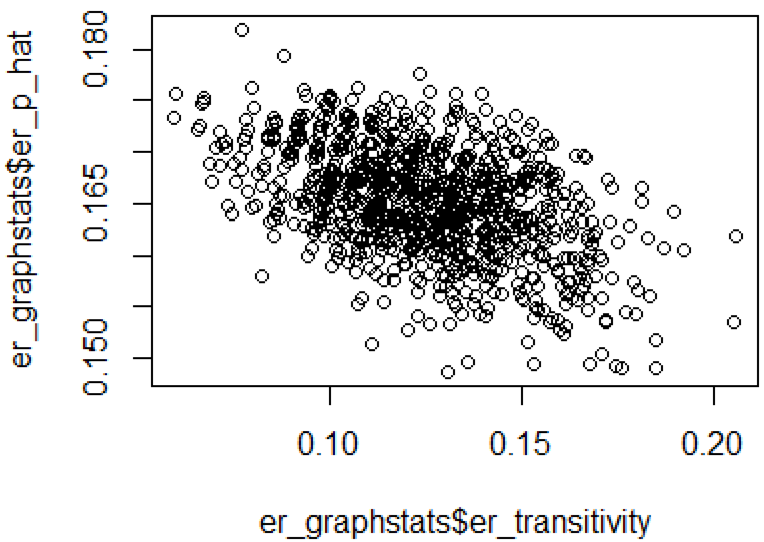
For sparse matrices, use inverse of leading eigenvalue of Hashimoto nonbacktracking matrix [due to Newman et al], also extracted from DSM (most DSM's are sparse).

```
> library(igraph)
> vertices=1000
> edges=2500
> ergraph<-sample_gnm(vertices,edges,directed = FALSE, loops =FALSE)
> A<-as_adjacency_matrix(ergraph)
> A_eigen<-eigen(A, symmetric = TRUE,only.values=TRUE)[1]
> I<-diag(vertices)
> D<-diag(degree(ergraph))
> B<-I-D
> Z<-matrix(0,nrow=vertices,ncol=vertices)
> H<-rbind(cbind(A,B),cbind(I,Z))
> H_eigen<-eigen(H, symmetric = TRUE,only.values=TRUE)[1]
> p_hat<-1/max(H_eigen$values)
> p_hat
[1] 0.1584883
```

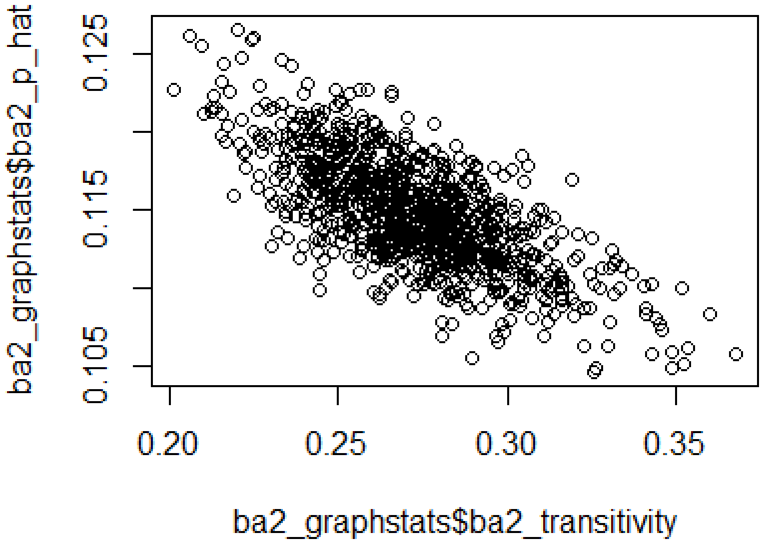
Source:
Radicchi, Filippo. "Predicting Percolation Thresholds in Networks." *Physical Review E*, vol. 91, no. 1, 2015, doi:10.1103/physreve.91.010801.

Main Findings I:
Clustering strongly correlates with increased network resilience, generally, lowering tipping point into connectivity

Erdos-Renyi Random Graphs

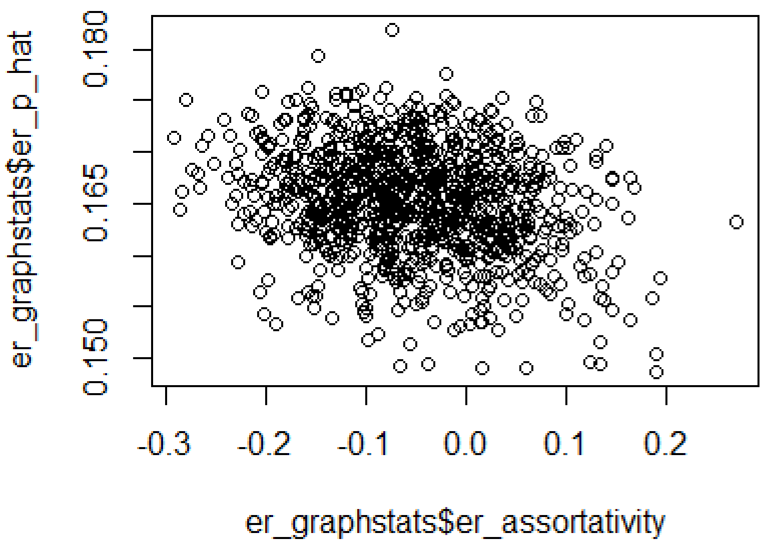


Barabasi-Albert Scale-Free Graphs, k=2

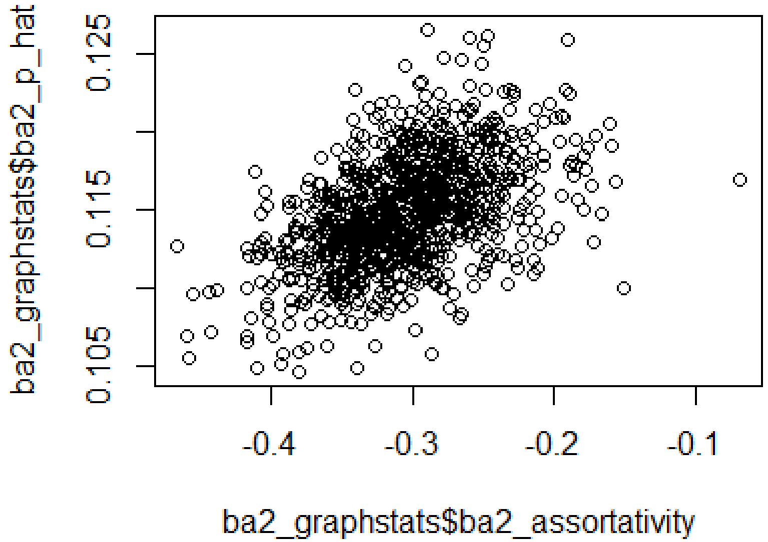


Main Findings II:
Degree correlation has weaker, mixed association with increased network resilience, generally.

Erdos-Renyi Random Graphs

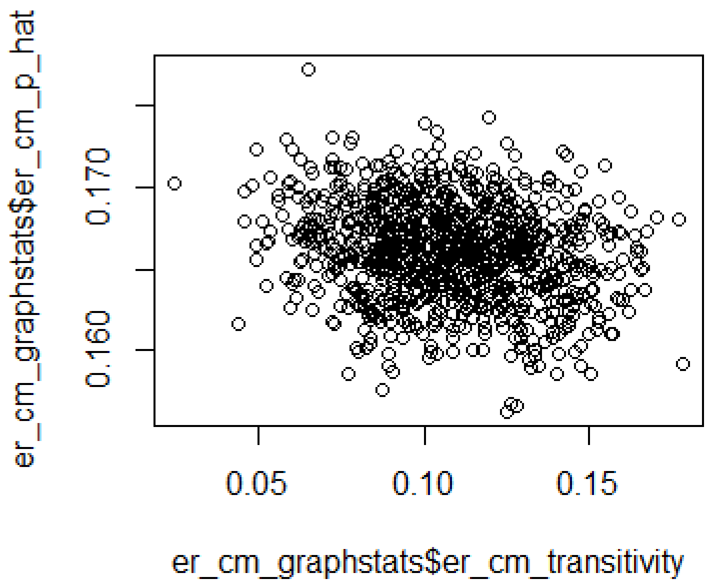


Barabasi-Albert Scale-Free Graphs, k=2

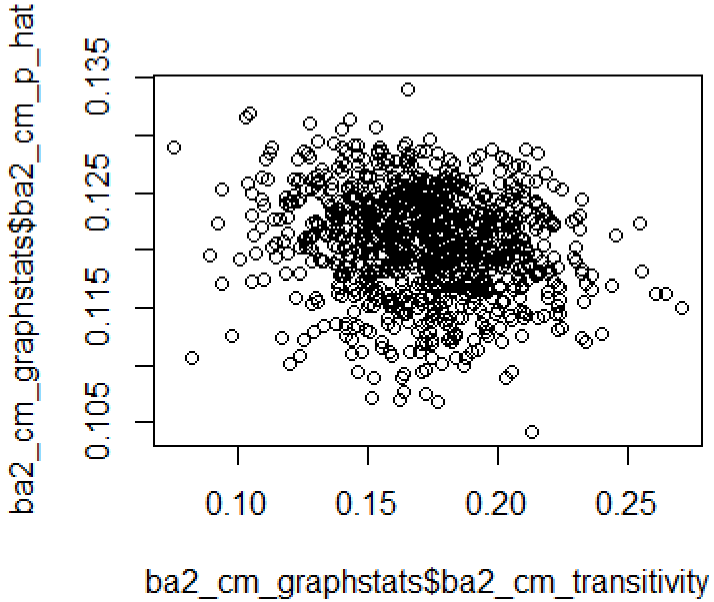


Main Findings III:
In contrast, clustering has almost no correlation with increased network resilience when keeping node degree constant, under the ‘configuration model’.

Erdos-Renyi Random Graphs

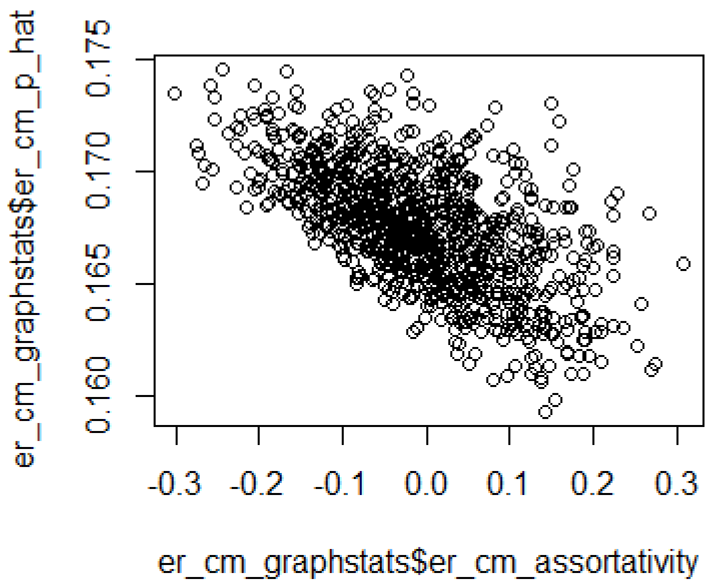


Barabasi-Albert Scale-Free Graphs, k=2

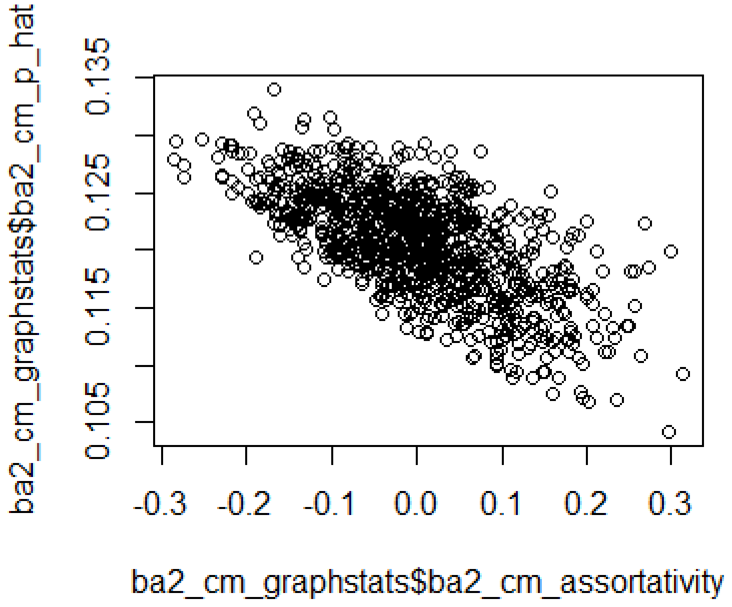


Main Findings IV:
Meanwhile, degree correlation has strong association with increased network resilience when keeping node degree constant, under the ‘configuration model’.

Erdos-Renyi Random Graphs



Barabasi-Albert Scale-Free Graphs, k=2



Main findings V:

Using the regularities observed above as heuristics, edge addition and rewiring rules can be derived. A consistent improvement of ~2% across all graph types is observed, between single ‘good’ and ‘bad’ edge additions / rewirings.

Effect of single worst rewiring, keeping degree distribution constant, on connectivity threshold.

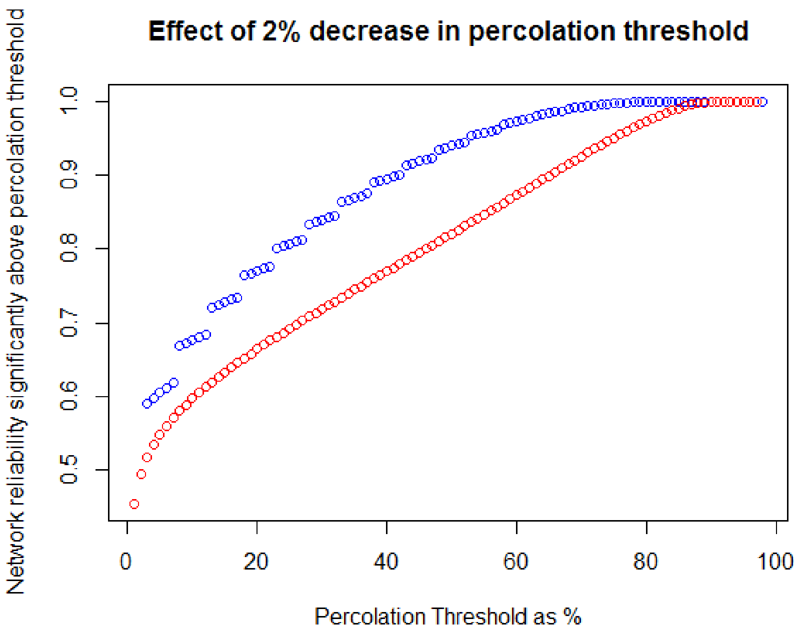
Effect of single best rewiring, keeping degree distribution constant, to maximally increase node degree correlation, on connectivity threshold.

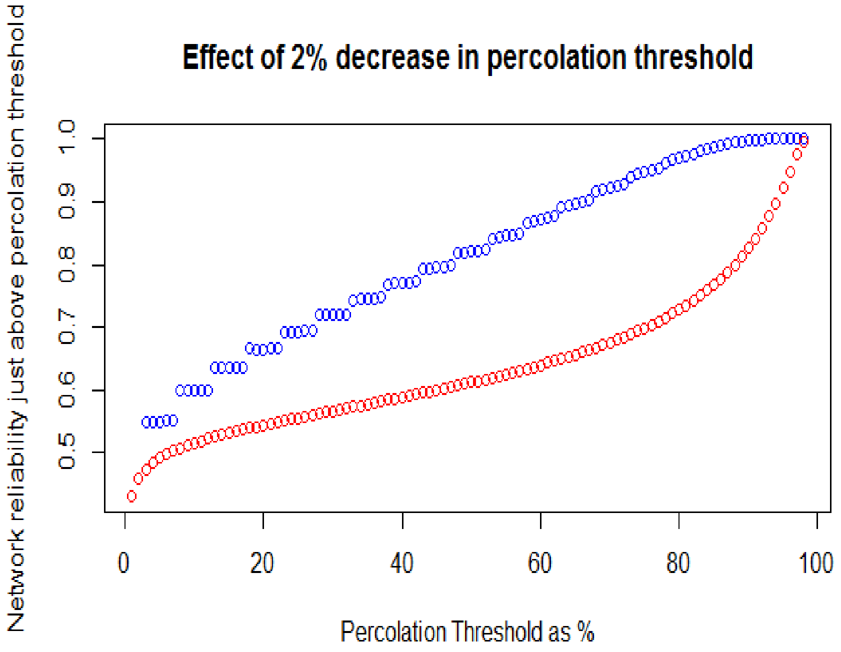
Effect of adding single best edge to network, to maximally increase network clustering coefficient, on connectivity threshold.

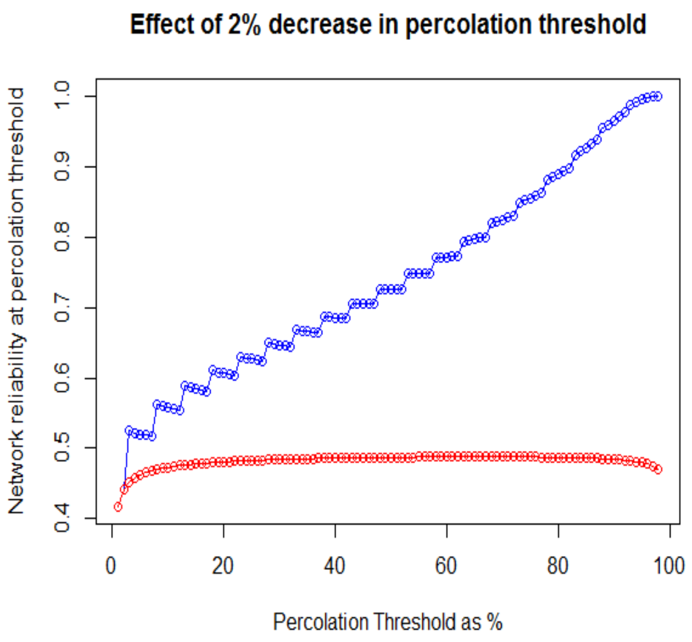
Effect of adding single bad edge, completing as few triangles as possible.

Effects of 2% decrease in percolation threshold

This effect is especially relevant when the system is performing (in terms of uniform node reliability) at or near the connectivity threshold. Overall network reliability calculations when system is performing comfortably above (3%), just above (1%) and at the percolation threshold, over a range of different thresholds, are as follows (improved network's performance is in blue):







Conclusions

Resilience can be modeled as an architectural property of networks

That capacity is best modeled via the notion of percolation threshold

Clustering (percent completed triangles) and homophily (node degree correlation) are best leverage points

Viewing architecture as communication, a network’s adaptive capacity to restore communications, after failure, is key

For socio-technical networks, leverage points are both necessary and useful

The Design Structure Matrix is a key tool in deploying node addition and rewiring rules

ERS
ENGINEERED RESILIENT SYSTEMS
DEPARTMENT OF DEFENSE

NDIA 2018 Systems Engineering Conference

Material here has been abstracted, summarized, or developed from a dissertation to be submitted to the George Washington University in partial fulfillment of the requirements for the Ph.D. in System Engineering.

NDIA
National Defense Industrial Association

Presenting co-author bio

BOB HILL is a PhD student in Systems Engineering at George Washington University. In addition to an undergraduate degree from Princeton, he has degrees in social science from Oxford and Cambridge, and in management and operations research from NYU and Columbia. With fifteen years of experience in quantitative finance, he is co-head of an algorithmic trading group for a Chicago-based broker-dealer. He can be reached at bhill42 at gwu.edu.