Detecting Evolving Patterns of Self-Organizing Networks by Flow Hierarchy Measurement

JIANXI LUO AND CHRISTOPHER L. MAGEE

Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Received June 7, 2010; revised August 15, 2010; accepted November 1, 2010

Hierarchies occur widely in evolving self-organizing ecological, biological, technological, and social networks, but detecting and comparing hierarchies is difficult. Here we present a metric and technique to quantitatively assess the extent to which self-organizing directed networks exhibit a flow hierarchy. Flow hierarchy is a commonly observed but theoretically overlooked form of hierarchy in networks. We show that the ecological, neurobiological, economic, and information processing networks are generally more hierarchical than their comparable random networks. We further discovered that hierarchy degree has increased over the course of the evolution of Linux kernels. Taken together, our results suggest that hierarchy is a central organizing feature of real-world evolving networks, and the measurement of hierarchy opens the way to understand the structural regimes and evolutionary patterns of self-organizing networks. Our measurement technique makes it possible to objectively compare hierarchies of different networks and of different evolutionary stages of a single network, and compare evolving patterns of different networks. It can be applied to various complex systems, which can be represented as directed networks.

Key Words: self-organizing networks; evolution pattern; flow hierarchy

INTRODUCTION

Complex systems of various kinds (social, biological, physical, technological, etc) frequently take the form of hierarchy [1, 2]. On one hand, hierarchy is one of the central structural schemes that an architect may use to manage complexities. Products, organizations, and other artifacts are often designed and managed hierarchically. On the other hand, hierarchies emerge and occur widely in self-organizing and evolutionary systems, such as food webs (ecological), neural networks (biological), open-source software (technological), and industrial production networks (economic), etc., which have no architect. In such cases, hierarchy is viewed as a natural emergent phenomenon and the consequence of evolutionary processes [2, 3].

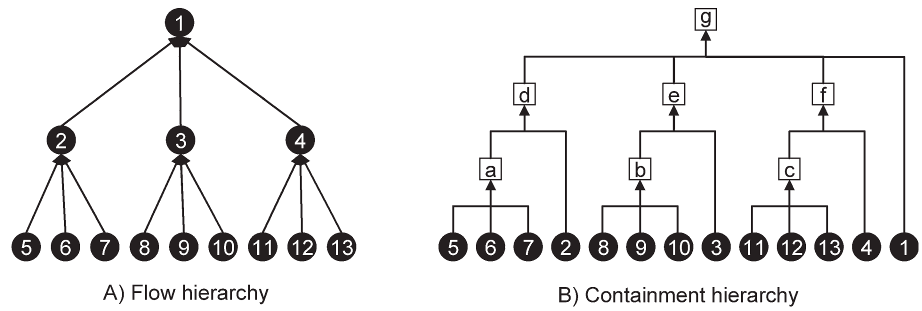
In complex self-organizing networks, hierarchy, like the well-studied "small world" phenomenon [4] and the power law of degree sequence [5, 6], is a global feature shared by various kinds of network systems (e.g., ecological, biological, social, and technological) [7–9]. It is important to

Additional Supporting Information may be found in the online version of this article.

Corresponding author: Jianxi Luo, Massachusetts Institute of Technology, 77 Massachusetts Avenue, E38-450, Cambridge, Massachusetts 02139. E-mail: luo@mit.edu

DOI 10.1002/cplx.20368
Published online in Wiley Online Library
(wileyonlinelibrary.com)

FIGURE 1



Alternative hierarchy representations of a single network.

understand the hierarchy in self-organizing networks, because as emergence it may reflect important information on the functional needs of or constraints on the entities and their relationships, which collectively form the network. However, detecting and comparing hierarchies is difficult in real-world networks, largely because first there are many types of hierarchy, and secondly hierarchy usually appears in impure forms in them [10, 11].

Hierarchy is a generic structure in which levels are asymmetrically ranked according to a specific type of relation. The ordering of levels, i.e., the rule of asymmetry, determines a hierarchy. Scholars interested in complex systems [1–3, 10] have paid attention to various types of relations existing between the elements that may determine a hierarchy and have described as many as four types of hierarchy in general [12]. By the logic construct for why an upper level is above a lower one, two types of hierarchies are useful for understanding the more specific case of network architectures: containment hierarchy and flow hierarchy.

A containment hierarchy is similar to the concepts of "nested hierarchy" [1, 10, 13] or "inclusion hierarchy" [12, 14] in which nodes are divided into groups that are further divided into subgroups of groups and so on over multiple levels. Containment hierarchy can be represented as a pure tree or dendrogram [11, 15] in which nodes that are closely connected [9, 11, 15–17], or have close equivalence measures [15, 18, 19], share lower common ancestors than more distantly connected or distinctly positioned nodes. A containment hierarchy can be found for both directed and undirected complex real-world networks.

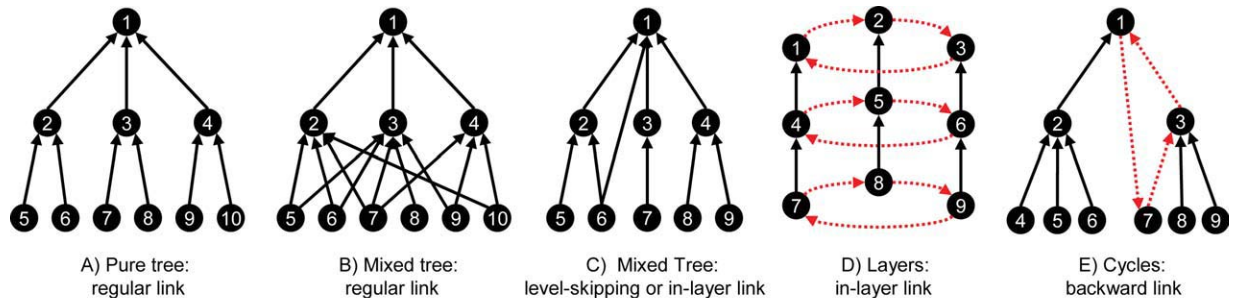
Flow hierarchy is only associated with directed networks but is observed in many evolving self-organizing networks such as food webs, neural networks, information processing networks, and industrial production networks. In many of these cases, the containment ordering criterion does not apply and the order of levels is essentially determined by the direction of the flows of resources essential to the network. Such flows are crucial because they provide necessary resources, for the entities to produce, reproduce, sustain (or remain in useful or necessary existence), and prosper. Via being connected by flows, the entities in such self-organizing systems coevolve and may self-organize into a flow hierarchy. For example, in food webs, it is energy that flows. In software networks, it is information that flows as subroutines feed parent routines. In the production network of firms, flow hierarchy arises when there is "persistent directionality in continuing flows of intermediate goods" [20] and flows of payments in a reverse direction. In production economies, firms coevolve in networks of flows.

A directed network may embed and exhibit both flow hierarchy and containment hierarchy. Figure 1 illustrates the distinction between these two alternative types of hierarchical representations for a single network. In Figure 1(A), the solid balls are the actual entities connected by the flows in the tree network. The flow hierarchy of the network is self-explanatory. Figure 1(B) is a containment hierarchy representation of the network in Figure 1(A) in which the squares stand for the subsystems (and the subsystems of subsystems) level by level, downward to the elementary entities (solid balls) of the network in Figure 1(A). Most software systems can be well represented in both ways: a flow hierarchy—a network of routines connected by directional information flows; as well as a containment hierarchy—a tree of the system, decomposed into a number of subsystems, each of which may be further sub-decomposed, recursively, until reaching the individual routines.

In some cases, only one type of hierarchy is appropriate. In the production network setting (as in food webs), the concept of flow hierarchy is meaningful, because firms are essentially involved and connected by the transaction flows to innovate and produce a coherent set of system

DOI 10.1002/cplx

FIGURE 2



Example networks. The dashed red links are involved in cycles.

products and coevolve. In such cases, a containment relation/structure is ambiguous.

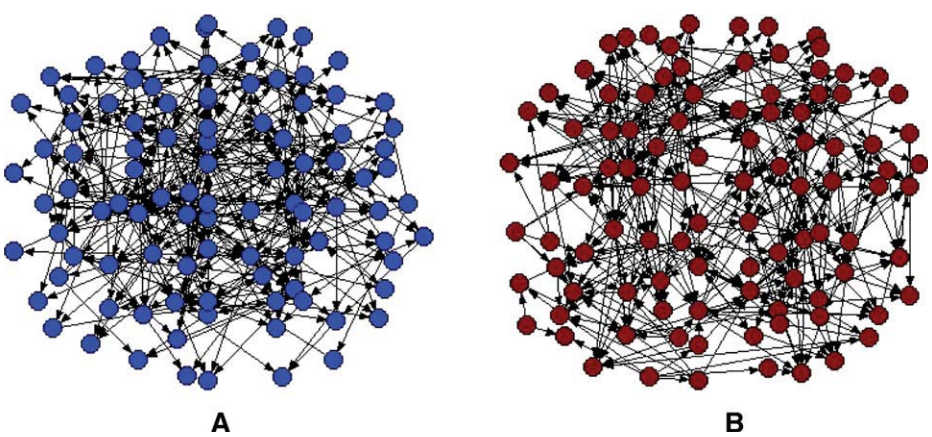
Imperfect Flow Hierarchy in Networks

Much of the recent interesting work [11, 15, 16, 21] on hierarchy in complex networks has been devoted to containment hierarchies. Although flow hierarchy also frequently occurs in various kinds of systems, it has been largely ignored. This article aims to promote awareness of flow hierarchy as an emergent property of complex self-organizing networks and as a lens to study and deepen our understanding on such networks.

The value of interpreting systems as flow hierarchies has not been fully exploited, partly because flow hierarchies usually do not appear in a pure form in complex self-organizing networks, such as food webs, neural networks, etc. Ideally, given a criterion used to link levels above and below, the links from a predefined lower level to its adjacent higher level are regarded as hierarchical. But we often observe links that skip levels, that connect between nodes on the same level, and that go in the backward direction. With all these irregularities aggregated in large complex networks, as well as the arbitrary nature of link type identification based on level assignment, flow hierarchies may become ambiguous and intractable. Figure 2 demonstrates several simple example networks, which embed and exhibit flow hierarchy to varied degrees.

Figure 2(A) is a pure tree. Each node is assigned not only a rank, but a single link to a higher up node. In Figure 2(B), some nodes have multiple inbound and outbound links. We call it a "mixed tree hierarchy." Both the pure tree and the mixed tree are strictly hierarchical because all the links regularly connect from a lower level to an adjacent higher level. In the network C in Figure 2, levels can no longer be uniquely defined. If node 2 and 6 are defined to be in the same layer, the link from node 6 to 2 can be viewed as an "in-layer link" and the link from 6 to 1 is a "regular link". However, if node 2 is predefined to one level higher

FIGURE 3



Random networks with the same size ($N = 100$, $L = 400$) but different hierarchy degrees. N is the number of nodes, L is the number of links.

DOI 10.1002/cplx

than node 6, then the link from 6 to 1 is a "level-skipping link." Identification of level-skipping links and in-layer links relies on the pre-identification of levels. In this case, the levels are not uniquely defined [22]. But at least all the links in Figure 2(C) follow a general asymmetrical direction, so this network can be regarded as hierarchical. In cases A, B, and C, there is strict asymmetric ordering of relationships.

Networks often exhibit layered structures [23], that is, level hierarchy [12], as shown in Figure 2(D). In this example, the links in cycles are symmetrical to each other and lose their global direction to some extent. However, if the nodes in the same directed cycle are presumed to be in a layer (then the links are "in-layer links"), the other links proceed in one direction from layer to layer. Thus, the network D in Figure 2 is not purely hierarchical but still has certain degree of hierarchy, that is, partially hierarchical. The example in Figure 2(E) simply shows how the emergence of a cycle may destroy the overall direction or asymmetry of a network. The examples in Figure 2 together indicate that cycles violate the directionality of a network, that is, the asymmetry in flows, which is the fundamental principle of flow hierarchy (i.e. things move in one general direction).

The networks in Figure 2 are simple, so we can intuitively observe and sense the different degrees of hierarchy embedded in them. When given more complex and larger networks, the identification of flow hierarchy can be difficult. Figure 3 visualizes two random networks with the same numbers of nodes (100) and links (400), but vastly different degrees of hierarchy embedded. It is not surprising but important that such visualization while useful does not allow one to objectively see significant differences in hierarchy between different networks. Our technique introduced in next paragraph will reveal the large difference in hierarchy between the two networks in Figure 3.

The Measurement of Flow Hierarchy

Centered on the concept of flow hierarchy and its core principle, network directionality, we present a hierarchy metric that detects and measures the extent to which all the local flows follow a holistic overall "underlying direction." The hierarchy metric is calculated as the percentage of links that retain their overall direction in the network, that is, the percentage of links that are not included in any cycle,

$$h = \frac{\sum_{i=1}^{L} e_i}{L} \tag{1}$$

where L is the number of links in the network and $e_i = 0$ if link i is in a cycle (1 otherwise). In weighted networks, the metric can be calculated as the ratio of the weights of the links, which are not included in any cycles over the total weight of all links,

$$h_w = \frac{\sum_{i=1}^{L} w_i e_i}{\sum_{i=1}^{L} w_i} \tag{2}$$

where w_i is the weight of link i. In this article, we will focus on unweighted networks.

TABLE 1

Hierarchy Degrees of the Example Networks in Figures 2 and 3

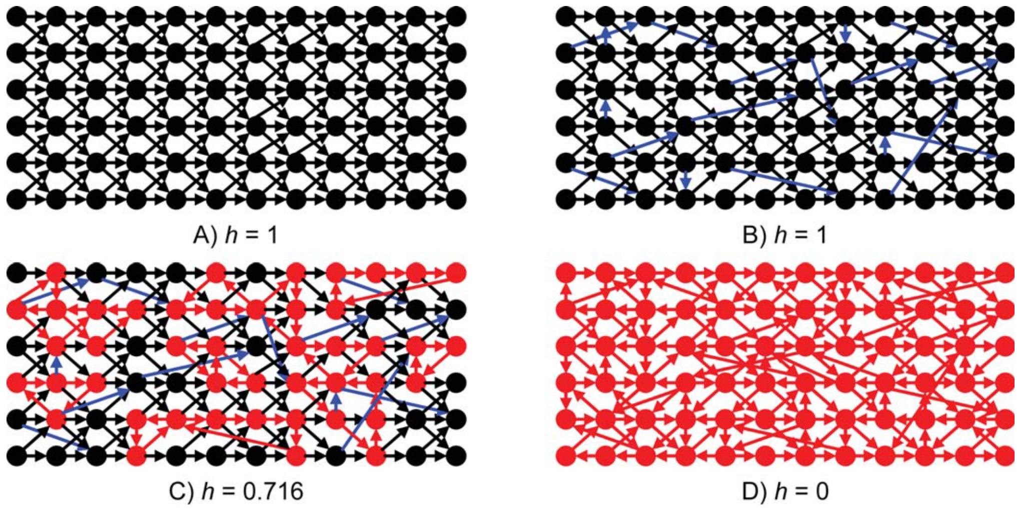
	Figure 2					Figure 3	
Networks	A	B	C	D	E	A	B
Hierarchy Degree	1	1	1	0.40	0.67	0.33	1

We applied the flow hierarchy metric to the simple networks shown in Figure 2 and the larger examples shown in Figure 3. The calculated hierarchy degrees (see Table 1) capture the same understanding based upon direct observations on the networks in Figure 2. In particular, the metric performs well in assessing layered hierarchy but other potential metrics do not. For the example of network D in Figure 2, if we alternatively count the portion of nodes rather than links, all the nodes are involved in cycles so the alternative hierarchy metric will be zero and fail to capture the sense of layered hierarchy of this network. In general this metric is unambiguous in differentiating the hierarchical components and non-hierarchical components [23]. It is also advantageous in its clarity and ease of computation, in comparison to other potential metrics (An assessment of alternative metrics is provided in the Supporting Information.). The Supporting Information, as well as network data and computer codes implementing the methods, can be found online at (http://www.mit.edu/~cmagee/luo_hierarchy/).

This metric of flow hierarchy potentially provides a way to characterize and detect different structural regimes of discrete systems with a potential direction, analogously to the different regimes of the continuous fluid flows. For example, the networks A and B in Figure 4 are strictly hierarchical (uni-directional) and similar to the "laminar flow" regime of fluid flows. In network C of Figure 4, some of the local flows (i.e., links) are involved in cycles (similar to eddies or vortexes of fluid flows). The system is no longer purely hierarchical, that is, partially hierarchical, and is in a "transitional flow" regime. In network D, all the flows are involved in cycles, so this case is analogous to the "turbulent" regime of fluid flows. Thus, as the Reynolds

DOI 10.1002/cplx

FIGURE 4



A network with 72 nodes and 176 directed links, oriented toward different directions in four scenarios. The links in blue color in (B) and (C) either skip levels or connect between nodes in the same level. Such links add complexity and difficulty in determining the levels and ranks, but do not destroy the overall network directionality, that is, flow hierarchy. The nodes and links colored in red are involved in cycles.

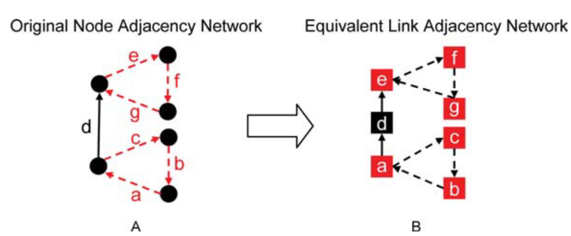
Number [24, 25] characterizes different flow regimes, such as laminar, turbulent, or transitional flow, the flow hierarchy metric also potentially characterizes the structural regimes of discrete network systems, such as production markets, food webs, and software.

To compute the flow hierarchy metric for large-scale complex networks, we use the following algorithm: First, we construct the link adjacency network and matrix for the original node adjacency network. For example, Figure 5 shows the link adjacency network transformed from and equivalent to the original node adjacency network. The seven squares in Figure 5(B) correspond to the seven links of the network of Figure 5(A), respectively.

We name the cell (i,j) in the link adjacency matrix x_{ij}. $x_{ij} = 1$ if and only if the end of link i is directly connected to the start of link j by a node. Otherwise, $x_{ij} = 0$. Second, we raise the link adjacency matrix's power p to find the link distance matrix M_d. We name the cell (i,j) in the link distance matrix d_{ij}. The d_{ij} is the distance from link i to j, defined as the minimum number of unique nodes which a uni-directed flow has to travel through from the end of link i to the start of link j. The d_{ij} is found as the value of the power, at which cell (i,j) of the power matrix M^p has a non-zero value for the first time. When $p = 1$, the power matrix M^1 is the same as the link adjacency matrix, so that if $x_{ij} = 1$, the distance from i to j is 1. If $x_{ij} = 0$, and $x^{[2]}_{ij} > 0$, then the distance is found as 2. And so forth. Consequently, the first power p for which the $x^{[p]}_{ij}$ element is non-zero gives the distance from i to j, that is, the value of d_{ij} in the link distance matrix M_d. Mathematically, $d_{ij} = \min p x^{[p]}_{ij} > 0$, for p from 1 to L, the total number of links (equal to the length of the longest possible cycle of links). We leave d_{ij} empty if the end of link i is neither directly nor indirectly connected to the start of link j. Note that alternative algorithms, such as depth-first search, can also be applied to find the link distance matrix.

Figure 6 illustrates the process to derive the link distance matrix for the example network in Figure 5. Given the final link distance matrix (at the bottom right corner of Figure 6), we are able to judge if a link is on any directed cycle by examining its main diagonal. If d_{ii} is empty, then link i is not involved in any cycle (i.e., $e_i = 1$, 0 otherwise). In this case, only d_{dd} is empty, and this agrees with our direct observation on Figure 5—only link d is not included in any cycle. Thus, the hierarchy degree is 1/7.

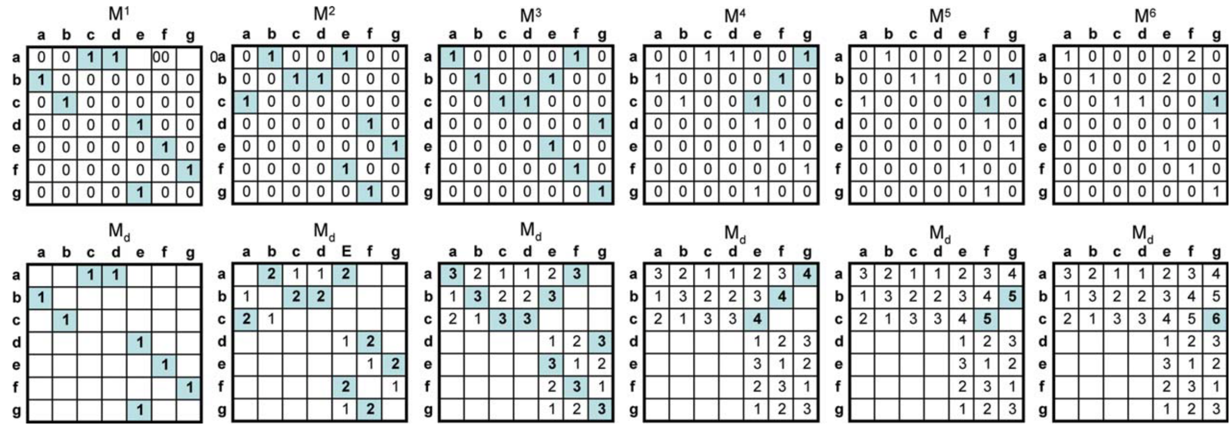
FIGURE 5



The link network equivalent to the original node network.

DOI 10.1002/cplx

FIGURE 6



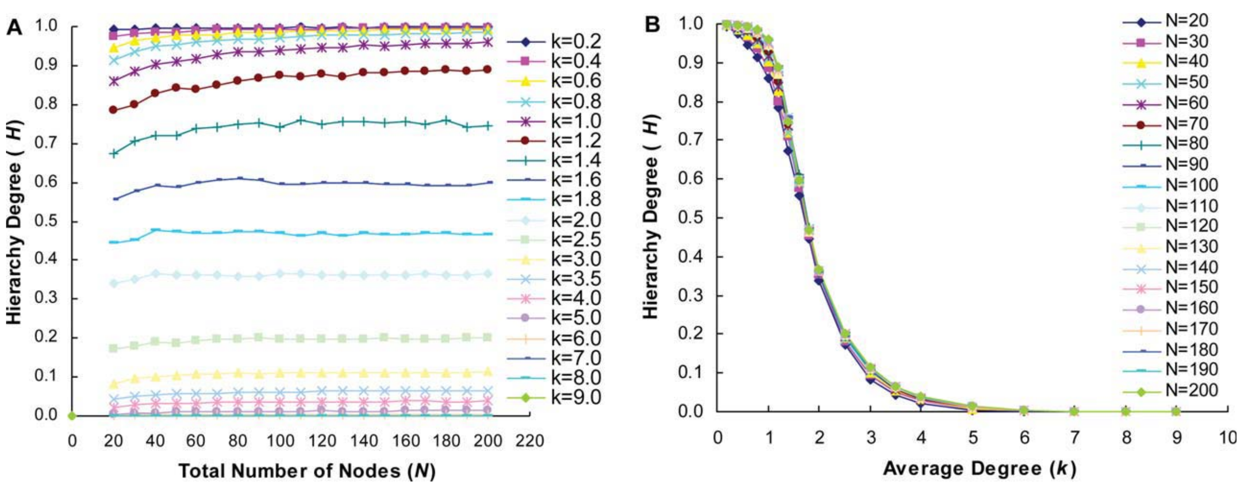
Deriving link distance matrix by raising power of link adjacency matrix. We pair M^p and the M_d with the state of knowledge after p steps. M^1 is the link adjacency matrix for the link network in Figure 5B and the original network in Figure 5A. The distance identified at each intermediate step is bold and its cell is shadowed. In particular, the values on the diagonal of the final M_d (after 6 steps in this case) give the length of the shortest link cycles in which each link is included.

RESULTS AND DISCUSSIONS

By definition, a pure random directed network embeds no hierarchy. However, the hierarchy degree is not necessarily zero for the networks created by existing random network models. We have examined hierarchy degrees of networks generated by a simple model similar to the "Poisson random network" [9]. Networks are constructed by assigning *L* directed links to randomly chosen pairs from *N* nodes. No multiple uni-directed links between a chosen pair and no self-links are allowed.

The directed Poisson random networks alone also exhibit important properties regarding hierarchy. Figure 7(A) shows that network size (*N)* has little influence on hierarchy degree (*h*), especially when *N* is sufficiently large. This agrees with our intuition that hierarchy is an architectural property independent of scale, and allows one to use ran-

FIGURE 7



Hierarchy degrees of randomly-generated directed networks. The value at each data point is the average of hierarchy degrees of 1,000 randomly-generated networks given the same N and k. Data points are connected by straight lines.

DOI 10.1002/cplx

TABLE 2

Hierarchy Degrees of Empirical Networks and Comparable Random Networks

Network	Type	N	L	k	h_{real}	$\hat{h}_{rand}$	$\hat{\sigma}_{rand}$
Bridge Brook Lake	Food Web	25	104	4.160	0.9809	0.0213	0.0338
NE US Shelf	Food Web	79	1378	17.443	0.8273	0	0
Japanese Automobile Sector	Production	679	2437	3.589	0.9988	0.0601	0.0114
Japanese Electronics Sector	Production	227	648	2.855	0.5957	0.1338	0.0310
C. elegans	Biological	280	2170	7.750	0.1171	0.0009	0.0018
D. melanogaster	Biological	107	301	2.813	0.3289	0.1308	0.0444
Darwin XNU-123.5	Software	646	4351	6.735	0.4872	0.0024	0.0021
Linux Kernel 1.1.70	Software	287	1385	4.826	0.8065	0.0159	0.0082

h_{real} is the hierarchy degree of the empirical network. $\hat{h}_{rand}$ and $\hat{\sigma}_{rand}$ are the average and standard deviation of the hierarchy degrees of an ensemble of 1,000 randomly-generated networks with the same N and L (or k) of the empirical network. We extracted the software call networks using the architecture analysis software Understand C++. In the call networks, a link from source code B to source code A exists if any function in A calls and relies on any function located in B. In the industrial production networks, a link from firm B to firm A exists if firm A procures any products from firm B.

dom networks with a relatively small N to estimate h of those with large N but the same k ($= L/N$). Figure 7(B) shows the increase of average degree (k) significantly decreases h. When k is at its minimum $1/N$, h will be exactly 1, because only one pair of nodes will be connected and one node is unambiguously above the other in this dyad flow hierarchy. When k is sufficiently high, h tends to zero because there are many cyclic pathways through which flow can travel back to its origin. A holistic direction does not exist among links in densely connected random networks. These results are of use when comparing h found from empirical networks.

We calculated the hierarchy metrics of a diverse set of empirical evolving self-organizing networks: the Bridge Brook Lake food web [26, 27] and the Northeastern US Shelf food web [28, 29], Japanese supplier-producer networks in automotive and electronics production sectors [30–32], two biological information-processing networks including the synaptic connections between neurons in the nematode worm *Caenorhabditis elegans* [33] and developmental transcription network of *Drosophila Melanogaster* [33], the call networks of the kernels of two operation system software, Linux [34] and Apple computer's Mac OS X (Darwin) [35]. We used the algorithm detailed above to compute the hierarchy degrees of these large-scale empirical networks. Then, hierarchy degrees of these empirical networks are compared with those of comparable Poisson random networks with the same numbers of nodes (N) and links (L). Results are listed in Table 2.

Domain-specific knowledge is needed to understand the detected difference in hierarchy degree of empirical networks of the same type. For the specific example of the two production networks, the automotive sector is significantly more hierarchical than the electronics sector (Table 2). This difference in hierarchy degree may imply and result from some important differences in the strategies and behaviors of individual firms and differences in the technological environments in the two production sectors.[1]

However, in the results (Table 2), there is no clear evidence to show that system types (e.g., biological vs. production) differentiate networks in terms of flow hierarchy. In general, from a network science perspective, the results show all of these typical empirical networks exhibit stronger hierarchical architectures than comparable random networks with the same sizes and average degrees, indicating flow hierarchy as a significant feature of real-world self-organizing networks.

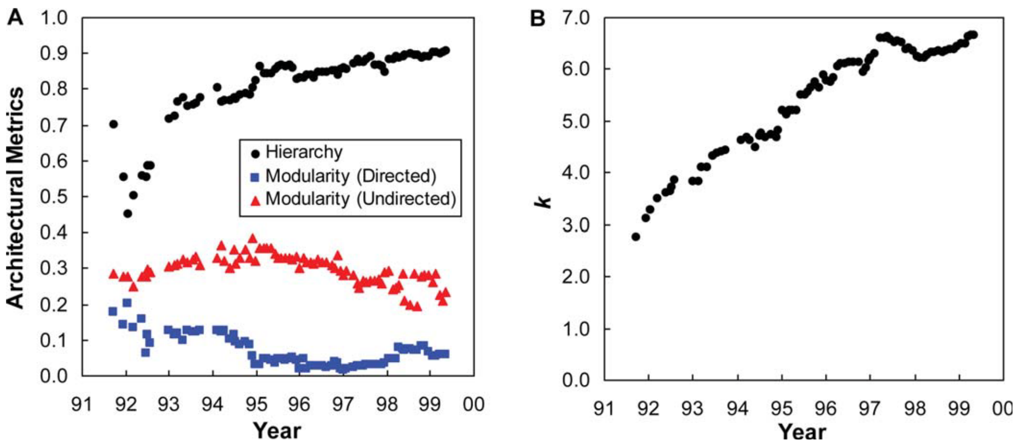
The flow hierarchy metric can also be used to quantitatively detect the evolving patterns of self-organizing networks. For containment hierarchy, Simon [1] hypothesized that hierarchy emerges inevitably through a wide variety of evolutionary processes because hierarchical structures are stable [1, 36]. However, quantitative evidence has not been reported previously, largely due to the lack of an appropriate measure. The hierarchy metric and technique in this article allows us to detect how flow hierarchy changes over the course of system evolution.

We calculated the flow hierarchy metrics of the call networks of various historical versions of the Linux kernel from its origin, version 0.01 to 2.3.0. The Linux kernel is an open source system developed by self-organized contributors around the world. As indicated in Figure 8(A),

[1] *Reference [32] attempts to explain why different industrial sectors may exhibit different degrees of hierarchy in the specific industrial and economic context.*

DOI 10.1002/cplx

FIGURE 8



Longitudinal evolution of Linux kernel. (A) Hierarchy degree and modularity. (B) Average degree. At least one data point is included for each month when there are multiple releases in a month. For some months, no data point is included, because there were no versions either released or available in the online archive. See Supporting Information for more details of the data and results.

the hierarchy degree of the Linux kernel has been generally increasing over its life cycle. The first version (0.01) was built and released by a single person. After that, many people contributed subroutines to the project, and thus hierarchy degree declined for a little while. During the most of its life as an open-source system, the hierarchy degree has increased as the self-organizing system grows, stabilizes, and matures. The observation of a general increase of k as the system evolves [Figure 8(B)] affirms the hierarchical tendency of this system since increases in k alone would work to decrease the hierarchical metric.

Network decompositions may reveal certain underlying architectures and interesting methods to detect modularity have been developed recently [11, 16, 19, 21, 37–39]. We also calculated the optimal modularity of the Linux kernel networks, using Newman's eigenvector-based algorithms for both undirected [38] and directed networks [39], and found unclear trends during the same period of time, if not slightly decreasing. No theoretical or observational indication has been found about how modularity of self-organizing networks should change in evolutionary processes. Compared to the flow hierarchy metric, the usefulness of modularity in terms of tracking the evolving patterns of self-organizing networks may be limited.

CONCLUSION

In general, this article explores a commonly observed but theoretically overlooked form of hierarchy in networks—flow hierarchy. Our measurement technique makes it possible to objectively compare hierarchies of different networks, detect the structural regime or evolutionary stages of a single network, and compare the evolving patterns of different networks. Our analysis shows that the ecological, neurobiological, economic, and information processing networks are generally more hierarchical than their comparable random networks. We further discovered that hierarchy degree has increased over the course of the evolution of Linux kernels. Taken together, the results may suggest that flow hierarchy is a central organizing feature of real-world evolving networks.

Our major purpose of this article is not to argue flow hierarchy must increase or decrease in the evolutionary course of a complex system, but to quantitatively examine flow hierarchy and show the power of the flow hierarchy metric to detect evolving patterns of self-organizing networks. This article is not intended as the final word on flow hierarchy, but a beginning of further and boarder research on it. Important questions, such as what flow hierarchy means to the functional performance of a network and how flow hierarchy emerges from the behaviors and interactions of individual network nodes, have not been answered.

We anticipate application of the hierarchy metric and measurement technique to more systems, such as ecological, biological, brain, and neural, social and technological systems, to help understand better their domain-specific architectures and evolutionary patterns. Like the contribution of Reynolds Number for the development of the overall field of fluid mechanics, the flow hierarchy metric may also potentially provide great value for designing and managing complex network systems, but further research is needed.

ACKNOWLEDGMENTS

We thank Daniel Whitney, Carliss Baldwin, Joel Moses, and Luis A. N. Amaral for insightful and stimulating discus-

DOI 10.1002/cplx

sions, Jennifer Dunne for providing the food web data, Ron Milo for providing the biological network data, and Kevin Groke for technical support to extract the call networks of software packages. We also thank MIT Portugal Program Engineering Systems Fundamentals Project for providing funding for this research.

REFERENCES

1. Simon, H. The architecture of complexity. Proc Am Phil Soc 1962, 106, 467–482.
2. Holland, J. Emergence: From Chaos to Order; Addison-Wesley: Reading, Massachusetts, 1998.
3. Anderson, P.W. More is different: Broken symmetry and the hierarchical nature of science. Science 1972, 177, 393–396.
4. Watts, D.J.; Strogatz, S. Collective dynamics of 'small-world' networks. Nature 1998, 393, 440–442
5. Price, D.J. de Solla. Networks of scientific papers. Science 1965, 149, 510–515.
6. Barabasi A.-L.; Albert, R. Emergence of scaling in random networks. Science 1999, 286, 509–512.
7. Albert, R.; Barabasi, A.-L. Statistical mechanics of complex networks. Rev Mod Phys 2002, 74, 47–97.
8. Strogatz, S.H. Exploring complex networks. Nature 2001, 410, 268–276.
9. Newman, M.E.J. The structure and function of complex networks. SIAM Rev 2003, 45, 167–256.
10. Ahl, V.; Allen, T.F.H. Hierarchy Theory: A Vision, Vocabulary and Epistemology; Columbia University Press: New York, 1996.
11. Clauset, A.; Moore, C.; Newman, M.E.J. Hierarchical structure and the prediction of missing links in networks. Nature 2008, 453, 98–101.
12. Lane, D. Hierarchy, complexity, society, In hierarchy in Natural and Social Sciences; Pumain, Denise Ed.; Springer-Verlag: New York, 2006; pp 81–119.
13. Alexander, C. Notes on the Synthesis of Form; Harvard University Press: Cambridge, MA, 1964.
14. Wilson, D. Forms of hierarchy: a selected bibliography. In: Hierarchical Structures; Whyte, L.L.; Wilson, A.G.; Wilson, D. Eds.; American Elsevier: New York, 1969; pp 287–314.
15. Wasserman, S.; Faust, K. Social Network Analysis; Cambridge University Press: Cambridge, 1994.
16. Sales-Pardo, M.; Guimera, R.; Moreira, A.A.; Amaral, L.A.N. Extracting the hierarchical organization of complex systems. Proc Natl Acad Sci USA 2007, 104, 15224–15229.
17. Ravasz, E.; Barabasi, A.-L. Hierarchical organization in complex networks. Phys Rev E 2003, 67, 02611.
18. Hsieh, M.H.; Magee, C.L. An algorithm and metric for network decomposition from similarity matrices: Application to positional analysis. Soc Networks 2008, 30, 146–158.
19. Hsieh, M.-H.; Magee, C.L. A new method for finding hierarchical subgroups from networks. Social Networks 2010, doi:10.1016/j.socnet.2010.03.005
20. White, H.C. Businesses mobilize production through markets: parametric modeling of path-dependent outcomes in oriented network flows. Complexity 2002, 8, 87–95.
21. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. Proc Natl Acad Sci USA 2002, 99, 7821–7826.
22. Nakano, T.; White, D.R. Network structures in industrial pricing: the effect of emergent roles in Tokyo supplier-chain hierarchies. Structure Dynamics: eJournal of Anthropological and Related Sciences 2007, 2, 1–23.
23. Moses, J. Three design methodologies, their associated organizational structures and their relationship to various fields. Proceedings of Engineering Systems Symposium, MIT, Cambridge, Massachusetts, USA, 2004. Available at http://esd.mit.edu/symposium/pdfs/papers/moses.pdf.
24. Reynolds, O. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. Philos Trans R Soc London 1883, 174, 935–982.
25. Rott, N. Note on the history of the Reynolds number. Ann Rev Fluid Mech 1990, 22, 1–11.
26. Havens, K. Scale and structure in natural food webs. Science 1992, 257, 1107–1109.
27. Dunne, J.A.; Williams, R.J.; Martinez, N.D. Network structure and biodiversity loss in food webs: robustness increases with connectance. Ecology Lett 2002, 5, 558–567.
28. Link, J. Does food web theory work for marine ecosystems. Mar Ecol Prog Ser 2002, 230, 1–9.
29. Dunne, J.A.; Williams, R.J.; Martinez, N.D. Network structure and robustness of marine food webs. Mar Ecol Prog Ser 2004, 230, 291–302.
30. Dodwell Marketing Consultants. The Structure of Japanese Auto Parts Industry; Dodwell Marketing Consultants: Tokyo, 1993.
31. Dodwell Marketing Consultants. The Structure of Japanese Electronics Industry; Dodwell Marketing Consultants: Tokyo, 1993.
32. Luo, J. Hierarchy in industry architecture: Transaction strategy under technological constraints. Doctoral Thesis 2010, Massachusetts Institute of Technology, Cambridge, MA.
33. Milo, R.; Itzkovitz, S.; Kashtan, N.; Levitt, R.; Shen-Orr, S.; Ayzenshtat, I.; Sheffer, M.; Alon, U. Superfamilies of Evolved and Designed Networks. Science 2004, 303, 1538–1542.
34. Linux Kernel Organization. Inc. http://www.kernel.org/. Accessed on 31 July 2008.
35. Apple, Inc. http://www.opensource.apple.com/darwinsource/10.0/. Accessed on 31 July 2008.
36. Agre, P. Hierarchy and history in Simon's "Architecture of complexity". J Learning Sci 2003, 12, 413–426.
37. Guimera, R.; Sales-Pardo, M.; Amaral, L.A.N. Module identification in bipartite and directed networks. Phys Rev E 2007, 76, 1–8.
38. Newman, M.E.J. Modularity and community structure in networks. Proc Natl Acad Sci USA. 2006, 103, 8577–8582.
39. Leicht, E.A.; Newman, M.E.J. Community structure in directed networks. Phys Rev Lett 2008, 100, 118703.

DOI 10.1002/cplx