



CHARACTERIZING THE STRUCTURAL QUALITY OF GENERAL COMPLEX SOFTWARE NETWORKS

JING LIU*, JINHU LÜ†, KEQING HE‡ and BING LI§

*State Key Laboratory of Software Engineering,
Wuhan University, Wuhan 430072, P.R. China*

**j_liu@whu.edu.cn*

‡hekeqing@public.whu.edu.cn

§libingyk@public.whu.edu.cn

*†LSC, Institute of Systems Science,
Academy of Mathematics and Systems Science,
Chinese Academy of Sciences,
Beijing 100080, P.R. China*

†jhlu@iss.ac.cn

CHI K. TSE

*Department of Electronic and Information Engineering,
Hong Kong Polytechnic University,
Hong Kong, P.R. China*

encktse@polyu.edu.hk

Received February 25, 2007; Revised April 17, 2007

Software systems can be modeled as complex networks in which software components are abstract nodes and their interactions are abstract edges. This paper attempts to characterize the structural quality of complex software networks. We propose to use a novel statistical measure, called *average propagation ratio*, to characterize the structural quality of general complex software networks, such as software adaptivity and maintainability. Several real-world complex software networks are analyzed in some depth to demonstrate the application of average propagation ratios. Furthermore, we investigate the key factors that determine the average propagation ratios of general complex software networks, resulting in a set of guiding principles that can be used in practical network design for improving the structural quality of complex software systems.

Keywords: Complex networks; software systems; structural quality.

1. Introduction

Over the last decade, complex networks have been intensively studied within the mathematics, physics, biological science, nonlinear science, information science, and engineering communities [Barabási & Albert, 1999; Challet & Lombardoni,

2004; De Moura *et al.*, 2005; MacCormack *et al.*, 2006; Myers, 2003; Potanin *et al.*, 2005; Ulrich, 1995; Valverde & Solé, 2003, 2005; Watts & Strogatz, 1998; Zhou *et al.*, 2006]. Real-world software systems can be regarded as complex networks, in which software components, such as

objects, classes, packages, subsystems or modules, are abstract nodes and the relationships (or interactions) between components are abstract edges. It has been shown that many typical software networks are scale-free or small-world [Challet & Lombardoni, 2004; De Moura *et al.*, 2005; Liu *et al.*, 2006; MacCormack *et al.*, 2006; Myers, 2003; Potanin *et al.*, 2005; Valverde & Solé, 2003].

Of particular interest in software engineering are object-oriented software systems [Liu *et al.*, 2006; Potanin *et al.*, 2005], where nodes represent classes and links represent their relationships such as association, inheritance and dependency. In a previous work [Liu *et al.*, 2006], we have found that, in addition to the power-law distributions of in-degree and out-degree, the function distribution among classes also obeys power-law distribution. Moreover, the strength of the function within a class is closely related to the out-degree of its counterpart node in an object-oriented network.

For a given software product, there exist a number of different architectures that realize the same function. Different architectures may give different properties, such as product quality, reliability, robustness, physical size, etc. [Liu *et al.*, 2006; Myers, 2003]. It has been shown that most software systems obey the same rules of degree and function distributions. Several interesting models were proposed to explain the emergent mechanisms of the specific software structures, such as the *duplication based* model [Valverde & Solé, 2003] and the *refactoring based* model of software evolution [Myers, 2003]. However, the real reason why the disparate software networks have similar network structures remains unclear. As software development is a continuous process which requires collective efforts of many software developers and engineers, frequent changes during the development stage are inevitable. However, due to the interdependency between classes, the change in one often requires corresponding modification of several others. It would be desirable to keep the change as “local” as possible, i.e. not involving a wide range of classes.

Recently, MacCormack *et al.* [2006] introduced the concept of *propagation cost* to measure the average influence of components on the whole system for some specific software networks. Basically, the propagation cost of a software system should be kept as low as possible. That is, the influence of any performed change should be limited to a range of components which are as small as

possible. A similar quantity, called *failure propagation basin*, was also proposed for the study of bug propagation and debugging efficiency [Challet & Lombardoni, 2004]. The failure propagation basin measures the potential influence of a faulty node and is defined as the number of all reachable nodes from the faulty node. Furthermore, the *debugging basin* is defined as the maximum number of software components that have to be inspected in order to locate the faulty node. In other words, the debugging basin should include all the nodes that can reach the node where software failure occurs. Therefore, in some sense, such measure is an indicator of the ease of bug propagation and debugging. These works tell us that software structure is closely related to both the extensibility and the testability of software, i.e. software maintainability. The better the software structure, the better software maintainability and the lower software development cost.

As software systems become more and more complex now, it is of great practical interest to address the basic question of how structural quality can be quantified for software networks. In this paper, a novel statistical characteristic, to which we refer to as *average propagation ratio*, is proposed to characterize the structural quality of general complex software networks, such as adaptivity and maintainability. In software design, it would be desirable to keep the average propagation ratio to be as small as possible, in order to reduce the cost or influence of change propagation. Clearly, the average propagation ratio is related to the number of network nodes, the number of network edges, and network distribution. We will also investigate the key factors that determine the average propagation ratios of general complex software networks. Our study will provide useful guiding principles for improving the structural quality of real-world software systems.

Our results reveal several important properties. First, the dependence between the average propagation ratios and the network diameters (or average shortest lengths) gradually reduces as the number of edges increases. Second, the dependence between the average propagation ratios and the differences between the maximum and minimum out- (or in- or both) degrees initially reduces as the number of edges increases, but gradually increases as the number of edges gets larger. Third, a star structure has much lower average propagation ratios

and a chain structure has much higher average propagation ratios. These observations are very useful for designing software network structures and ultimately for improving the structural quality of general complex software networks.

The rest of this paper is organized as follows. Section 2 introduces the concept of *average propagation ratio* and its use in characterizing the structural quality of general complex software networks. In Sec. 3, several real-world complex software networks and general complex software networks with different distributions are studied in detail. Finally, we give some conclusions in Sec. 4.

2. Preliminaries

In this section, we introduce the concept of *average propagation ratio*, which is relevant to the characterization of the structural quality of general complex software networks.

In an object-oriented (OO) software system, an element is a class, and the relationships between classes are inheritance, association, dependency, etc. We model an OO software system as a directed graph, which is denoted as $\mathbf{G} = \langle \mathbf{N}, \mathbf{E} \rangle$, where \mathbf{N} is the set of all classes of the specific OO software system, and \mathbf{E} is the set of all relationships between classes. Here, $\langle n_i, n_j \rangle \in \mathbf{E}$ means that class n_i either inherits from, or is associated with, or depends on class n_j . Denote \mathbf{R}_{ik} as the set of all reachable nodes originated from node n_i within a distance k . Then, the *average propagation ratio* of a specific complex software network within a distance k is defined as

$$PC_k = \frac{\sum_{i=1}^{|\mathbf{N}|} |\mathbf{R}_{ik}|}{|\mathbf{N}|^2} \quad (1)$$

where $|\mathbf{N}|$ denotes the number of classes.

Let the network diameter be D . According to the above definition of average propagation ratio, if $k > D$, then $PC_k = PC$. Therefore, the average propagation ratio of a specific complex software network can be defined as

$$PC = \frac{\sum_{i=1}^{|\mathbf{N}|} \left| \bigcup_{k=0}^{\infty} \mathbf{R}_{ik} \right|}{|\mathbf{N}|^2} = \frac{\sum_{i=1}^{|\mathbf{N}|} \left| \bigcup_{k=0}^D \mathbf{R}_{ik} \right|}{|\mathbf{N}|^2}. \quad (2)$$

Moreover, from (1) and (2), if $k_1 > k_2$, then $PC_{k_1} > PC_{k_2}$. This is because $\mathbf{R}_{ik_1} \supseteq \mathbf{R}_{ik_2}$, where \mathbf{R}_{ik_1} and \mathbf{R}_{ik_2} are the sets of all reachable nodes

originated from node n_i within the distances k_1 and k_2 , respectively.

In software engineering, average propagation ratios can be used to measure the degree of influence of a change in a component on other components in a specific software system, either directly or indirectly through a chain of dependencies. The proposed average propagation ratio can be used as a measure of software structural quality as far as software maintainability is concerned. Small value of average propagation ratio indicates that the measured software is apt to be changed and debugged. It is desirable to keep the average propagation ratios as small as possible for a specific software system, so that any change can be implemented with a minimum possible set of modifications.

Figure 1 shows the average propagation ratios of a simple network for $k = 1$ and $k = 2$. Here, the network has five nodes, i.e. $\mathbf{N} = \{A, B, C, D, E\}$. Clearly, class A depends directly on classes B, C and D. Thus any change in classes B, C or D will have potential influence on class A, i.e. $\mathbf{R}_{A1} = \{A, B, C, D\}$. Similarly, class C depends on classes D and E, i.e. $\mathbf{R}_{C1} = \{C, D, E\}$; classes B, D and E do not have output edges, i.e. $\mathbf{R}_{B1} = \{B\}$, $\mathbf{R}_{D1} = \{D\}$, and $\mathbf{R}_{E1} = \{E\}$. Therefore, the average propagation ratio within distance 1 is $PC_1 = (|\{A, B, C, D\}| + |\{B\}| + |\{C, D, E\}| + |\{D\}| + |\{E\}|) / (5 * 5) = 0.4$.

For propagation distance of 2, any change in class E has a direct and indirect influence on classes C and A, respectively. Thus, $\mathbf{R}_{A2} = \{A, B, C, D, E\}$. Since there is no path with length 2 other than A-C-E, the reachable sets of other classes are unchanged. Hence, $PC_2 = 0.44$. Furthermore, the diameter of this network is 2 and $PC_k = PC_2$ for $k > 2$.

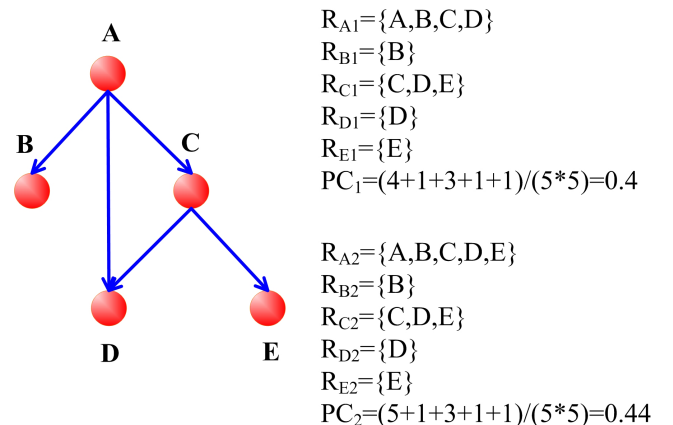


Fig. 1. Illustration of average propagation ratios of a simple network.

3. Characterization of the Structural Quality of Complex Software Networks

In this section we apply the afore-described average propagation ratio to measure the structural quality of general complex software networks.

3.1. Statistical parameters of selected software examples

As real-world software networks become more and more complex, structural quality will be an important attribute for ease of design and maintenance. Some recent results have shown that many practical software networks are small-world or scale-free [De Moura *et al.*, 2005; Liu *et al.*, 2006; Lü *et al.*, 2004]. In the foregoing, we have used the average propagation ratio to describe the structural quality of general complex software networks. We have pointed out that it is desirable to keep the average propagation ratio as small as possible for a software network.

In the following, we will analyze the relationships between the traditional statistical characteristics and the average propagation ratios of several representative OO software systems, including Eclipse, J2sdk, Azureus, Tomcat, G3d, Nodepad++. Table 1 shows the statistical characteristics of these software networks, where [Valverde & Solé, 2003]

Num = number of nodes,

L = number of edges,

d = average shortest length,

D = diameter,

C = clustering coefficient,

C_{rand} = clustering coefficient of the corresponding random network,

K = average degree,

In = power-law exponent of in-degree distribution,

Out = power-law exponent of out-degree distribution,

PC = average propagation ratio,

PC_{rand} = average propagation ratio of the corresponding random network.

Although the six selected software networks have sizes ranging from 332 nodes to 7415 nodes, the average shortest lengths are all within 10. Moreover, the network sizes and the clustering coefficients are much larger than those of random networks of comparable sizes. Furthermore, the distributions of in-degree and out-degree resemble power-law distribution, with exponents varying from 1.93 to 3.1. Therefore, the six software networks have typical small-world and scale-free characteristics. Also, the average propagation ratios of the six software networks are much lower than those of the random networks of comparable sizes.

3.2. Factors affecting average propagation ratios

Since the average propagation ratios are closely related to the network structure, a natural question to ask is: *what are the key factors that determine the average propagation ratios of complex software networks?* In this subsection, we will further investigate the relationships between the average propagation ratios and several statistical parameters.

Our experiment begins with generating three network structures with different degree distributions, namely, random, exponential and power law. We then calculate the average propagation ratios of these networks. Our results show that the average propagation ratios are very sensitive to the number of edges of the software networks. Figure 2 displays the average propagation ratios versus the number of edges for random networks with 1000 nodes, where PC_k shown in the legend represents the average propagation ratio for distance k . We clearly observe

Table 1. Statistical parameters of selected object-oriented software networks.

	Num	L	d	D	C	C_{rand}	K	In	Out	PC	PC_{rand}
Eclipse	7415	37884	9.722	32	0.338	0.000689	10.218	2	2.9	0.087	
J2sdk	5420	25952	5.915	17	0.48	0.000883	9.576	1.99	1.93	0.173	
Azureus	2413	9457	6.985	19	0.322	0.00162	7.838	1.94	2.7	0.0485	0.957
Tomcat	1533	3876	3.853	11	0.318	0.00165	5.057	2.3	3.1	0.0288	0.803
G3d	563	871	2.285	7	0.115	0.00275	3.094	2.2	2.8	0.0141	0.376
Noteepad++	332	468	1.889	6	0.112	0.00425	2.819	2.3	2.3	0.0121	0.26

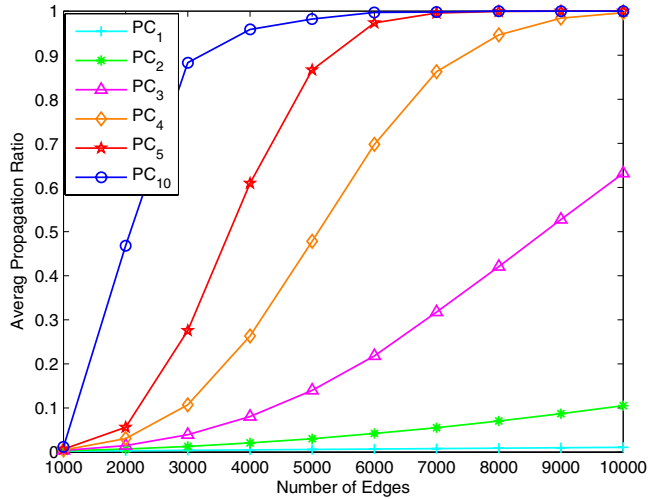


Fig. 2. Average propagation ratios of random networks with 1000 nodes. PC_1 , PC_2 , PC_3 , PC_4 , PC_5 and PC_{10} refer to the average propagation ratios within distances 1, 2, 3, 4, 5 and 10.

that the average propagation ratios of random networks increase with the number of edges for a given number of nodes. Moreover, the average propagation ratios are generally smaller for shorter k .

In the following, we study the relationship between the average propagation ratios and the distribution for a network with fixed numbers of nodes and edges. In this case, we generate various networks with different degree distributions, namely, random networks, exponential networks, and power law networks. In all numerical experiments, the number of nodes is 1,000 and the number of edges varies from 1,000 to 10,000. All characteristic parameters have been obtained from averaging over 100 independent experiments. Figure 3 shows the average propagation ratios of random networks, scale-free networks, and the exponential distribution networks. From Fig. 3, the following observations can be made:

- (1) Average propagation ratios generally increase with the number of edges.
- (2) Average propagation ratios are very dependent upon the network distribution.
- (3) There is no observable general relationship between the average propagation ratios and the types of the network distribution.
- (4) Average propagation ratios rapidly approach 1 for most network distributions.

Table 2 shows the average propagation ratios of complex software networks with 1000 nodes.

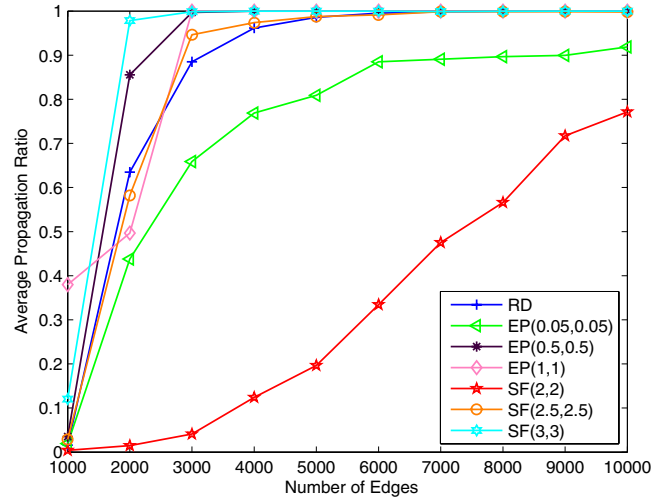


Fig. 3. Average propagation ratios of random networks, scale-free networks, and exponential distributed networks with 1000 nodes. RD denotes random distribution. $EP(i, i)$ for $i = 0.05, 0.5, 1$ denotes an exponential network with exponents of in-degree and out-degree distributions equal to i . $SF(i, i)$ for $i = 2, 2.5, 3$ denotes a scale-free network with exponents of in-degree and out-degree distributions equal to i .

Comparison is made between random distribution (RD), exponential distribution (EP), and scale-free distribution (SF). Distinction is also made between in-degree distribution and out-degree distribution when we consider EP and SF distributions. For instance, $EP(i, j)$ refers to exponential distribution with the exponent of the in-degree distribution being i and that of the out-degree distribution being j as shown in Tables 2–7. Likewise, we have $SF(i, j)$ denoting scale-free distribution with the exponent of the in-degree distribution being i and that of the out-degree distribution being j as shown in Tables 2–7. Furthermore, Tables 3 to 7 display the difference between the maximum and minimum out degrees, the difference between the maximum and minimum in-degrees, the difference between the maximum and minimum degrees, the average shortest lengths, and the diameter, respectively.

Network structure is directly characterized by the statistical parameters described earlier, e.g. average shortest lengths, network diameters, number of edges, etc. In order to have a better understanding of how average propagation ratios are related to these conventional statistical parameters, it is of interest to know the degree of correlation between the average propagation ratios of a complex software network with the statistical parameters of the network. Figure 4 reveals

Table 2. Average propagation ratios of software networks with 1000 nodes. RD: random distribution. EP(i, j): exponential network with exponent of in-degree distribution equal to i and exponent of out-degree distribution equal to j . SF(i, j): scale-free network with exponent of in-degree distribution equal to i and exponent of out-degree distribution equal to j .

PC	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
RD	0.0157	0.6349	0.8849	0.9617	0.9859	0.9952	0.9983	0.9994	0.9998	0.9999
SF(2,2)	0.0042	0.0149	0.0409	0.1241	0.1964	0.3343	0.4752	0.5659	0.7173	0.7713
SF(2,2.5)	0.0053	0.0717	0.3392	0.4788	0.6869	0.7319	0.8891	0.9413	0.9095	0.9544
SF(2,3)	0.0055	0.1334	0.3568	0.675	0.8561	0.9286	0.9665	0.9816	0.9862	0.9904
SF(2.5,2)	0.0062	0.0634	0.3425	0.5209	0.8079	0.8708	0.9379	0.9863	0.9827	0.9946
SF(2.5,2.5)	0.0287	0.5820	0.9462	0.9739	0.9874	0.9915	0.9986	0.9984	0.9987	0.998
SF(2.5,3)	0.0413	0.8880	0.9937	0.9999	0.9964	0.9990	0.9998	0.9986	0.9991	0.9996
SF(3,2)	0.0054	0.1175	0.4386	0.7824	0.9304	0.9713	0.9932	0.9987	0.9991	0.9996
SF(3,2.5)	0.0374	0.8426	0.9808	0.9982	0.9996	0.9988	0.9981	0.9986	0.9996	0.9996
SF(3,3)	0.1201	0.9783	0.9986	0.9999	0.9998	1	0.9989	0.9996	1	1
EP(1,1)	0.3798	0.4965	1	1	1	1	1	1	1	1
EP(0.5,0.5)	0.0352	0.8556	0.9969	0.9999	1	1	1	1	1	1
EP(0.05,0.05)	0.0194	0.4379	0.6587	0.7687	0.8091	0.8851	0.8908	0.8967	0.8995	0.9187

Table 3. Differences between maximum and minimum out-degrees of networks with 1000 nodes. Notations are same as Table 2.

PC	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
RD	5.52	7.91	9.9	11.86	13.65	15.35	16.55	17.72	18.94	20.07
SF(2,2)	108.11	263.38	371.14	459.17	581.82	607.44	614.77	633.72	662.07	699.83
SF(2,2.5)	54.84	140.37	148.53	241.25	268.37	298.84	308.69	290.34	364.33	350.26
SF(2,3)	28.43	48.36	75.05	86.43	120.22	129.7	144.43	162.19	154.84	176.82
SF(2.5,2)	104.49	269.51	351.81	433.57	446.6	456.34	453.98	464.42	468.19	486.08
SF(2.5,2.5)	58.13	117.13	121.26	170.87	171.11	201.37	206.01	194.55	219.52	228.9
SF(2.5,3)	25.23	42.23	63.54	69.06	83.49	84.18	87.38	103.67	105.32	108.46
SF(3,2)	106.61	249.37	324.37	327.03	312.22	356.14	373.68	360.83	376.25	384.82
SF(3,2.5)	58.8	115.1	130.95	135.3	141	148.24	177.52	168.1	177.62	191.41
SF(3,3)	30.25	43.99	59.66	58.23	63.02	75.77	87.64	84.75	99.37	99.41
EP(1,1)	4.88	9.24	13.68	16.97	21.2	25.58	28.8	31.25	34.78	39.45
EP(0.5,0.5)	6.33	12.04	17.03	22.8	28.82	32.77	36.01	41.7	45.64	48.9
EP(0.05,0.05)	7.22	14.73	21.31	28.65	34.23	41.21	47.64	52.54	60.79	64.91

Table 4. Differences between maximum and minimum in-degrees of networks with 1000 nodes. Notations are same as Table 2.

PC	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
RD	5.63	7.83	9.93	11.83	13.4	15.23	16.71	17.7	19.28	19.86
SF(2,2)	105.48	273.73	415.24	512.72	530.79	620.34	663.23	736.61	793.45	830.76
SF(2,2.5)	101.72	275.46	475.74	623.39	609.1	709.43	811.94	824.61	850.33	818.16
SF(2,3)	107.76	263.28	460.68	591.56	644.85	709.53	709.96	789.41	840.87	865.1
SF(2.5,2)	62.03	129.08	164.43	219.7	269.17	321.51	383.7	394.33	479.33	523.04
SF(2.5,2.5)	64.6	151.22	176.21	283.76	288.77	320.26	364	422.78	463.75	448.89
SF(2.5,3)	59.55	115.69	167.71	229.67	254.44	319.73	342.91	404.24	469.13	487.23
SF(3,2)	22.52	48.28	71.12	84.39	118.37	140.25	175.39	178.44	204.15	248.59
SF(3,2.5)	25.91	51.63	90.9	90.73	122.87	130	180.76	211.7	196.86	256.07
SF(3,3)	26.88	54.25	73.99	105.72	120.01	144.95	182.21	194.33	203.53	229.38
EP(1,1)	4.89	9.44	13.41	18.66	24.38	28.37	34.58	38.11	43.66	50.04
EP(0.5,0.5)	6.26	12.09	17.37	22.84	29.93	34.22	39.78	49.01	52.95	59.61
EP(0.05,0.05)	7.76	14.37	22.27	29.31	36.47	44.32	52.35	58.69	65.33	72.27

Table 5. Differences between maximum and minimum degrees of networks with 1000 nodes. Notations are same as Table 2.

PC	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
RD	7.82	11.94	15.42	17.81	20.04	22.09	23.97	25.34	26.99	28.65
SF(2,2)	130.86	343.92	520.65	635.38	700.76	758.83	769.81	822.7	874.43	908.98
SF(2,2.5)	112.97	309.95	489.43	650.59	642.51	731.29	823.75	835.08	867.45	832.98
SF(2,3)	110.2	266.41	461.2	595.29	651	712.15	713.61	793.88	846.29	871.43
SF(2.5,2)	117.37	290.62	373.41	459.37	481.19	526.8	532.3	565.57	593.61	638.58
SF(2.5,2.5)	87.66	195.85	200.8	314.42	315.56	356.45	389.5	440.72	480.49	463.17
SF(2.5,3)	61.65	122.33	175.44	236.82	257.08	326.17	346.82	406.69	473.52	491.33
SF(3,2)	107.54	252.18	328.37	330.36	324.98	365.62	393.65	375.55	400.13	421.89
SF(3,2.5)	63.24	128.35	160.35	158.24	177.5	175.66	241.13	251.97	245.78	302.43
SF(3,3)	41.32	70.2	92.79	115.59	126.72	153.47	194.4	199.28	213.91	238.61
EP(1,1)	6.9	12.51	17.99	22.98	30.02	34.59	42.44	45.46	52.36	58.98
EP(0.5,0.5)	8.91	16.43	22.01	30.49	38.53	44.95	50.32	58.8	64.2	72.13
EP(0.05,0.05)	9.88	19.46	28.88	38	47.32	56.23	66.57	75.16	82.08	92.07

Table 6. Average shortest lengths of networks with 1000 nodes. Notations are same as Table 2.

PC	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
RD	8.447	9.040	6.232	5.097	4.462	4.055	3.767	3.555	3.392	3.257
SF(2,2)	2.079	2.432	2.875	3.623	3.805	4.295	5.031	4.781	5.217	4.642
SF(2,2.5)	2.7	5.132	6.647	6.533	5.932	5.914	5.433	4.991	4.655	4.315
SF(2,3)	3.153	7.659	7.689	7.379	6.388	5.914	5.191	4.749	4.471	4.118
SF(2.5,2)	2.891	5.245	6.501	6.433	6.721	5.907	5.501	4.903	4.45	4.376
SF(2.5,2.5)	7.903	11.78	8.089	6.515	5.305	4.694	4.328	4.008	3.823	3.665
SF(2.5,3)	11.185	12.432	7.365	5.749	4.909	4.180	4.081	3.853	3.656	3.513
SF(3,2)	3.224	6.761	8.275	7.285	6.246	5.489	4.846	4.404	4.146	3.903
SF(3,2.5)	10.221	11.77	7.416	5.661	4.854	4.365	4.063	3.812	3.621	3.493
SF(3,3)	21.814	10.898	6.498	5.283	4.597	4.418	3.902	3.678	3.056	3.37
EP(1,1)	33.931	10.19	6.346	5.155	4.514	4.11	3.849	3.628	3.457	3.34
EP(0.5,0.5)	12.337	9.690	6.935	5.358	4.743	4.333	3.973	3.773	3.586	3.445
EP(0.05,0.05)	9.181	9.031	6.554	5.496	4.876	4.515	4.194	3.946	3.785	3.652

Table 7. Diameters of networks with 1000 nodes. Notations are same as Table 2.

PC	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
RD	27.65	23.8	14.68	11.17	9.25	7.98	7.09	6.43	6.08	5.58
SF(2,2)	5.87	6.41	7.13	8.55	8.7	9.26	10.2	9.47	9.91	8.42
SF(2,2.5)	8.75	13.87	14.9	13.46	11.54	11.16	9.82	8.99	8.31	7.76
SF(2,3)	10.73	20.01	18.23	16.08	13.27	11.95	10.22	9.01	8.46	7.54
SF(2.5,2)	9.43	13.67	15.15	14.47	13.88	11.91	10.46	8.98	8.1	7.79
SF(2.5,2.5)	24.8	28.22	17.59	12.43	9.83	8.46	7.52	6.95	6.5	6.25
SF(2.5,3)	35.58	30.98	15.33	10.49	8.72	7.01	6.97	6.39	6.14	5.88
SF(3,2)	11.28	16.86	18.72	16.47	13.93	11.57	9.83	8.49	7.74	7.16
SF(3,2.5)	33.15	29	15.73	10.39	8.6	7.44	6.96	6.52	6.11	6
SF(3,3)	64.66	26.8	11.67	9.08	7.71	7.56	6.43	6.07	5.87	5.63
EP(1,1)	93.13	27.3	13.17	9.25	7.78	7.03	6.24	6.03	5.73	5.46
EP(0.5,0.5)	36.62	24.91	17.97	11.54	9.95	8.6	7.14	6.95	6.3	6.04
EP(0.05,0.05)	28.84	23.67	16.69	14.11	12.56	11.72	10.48	9.93	9.41	9.26

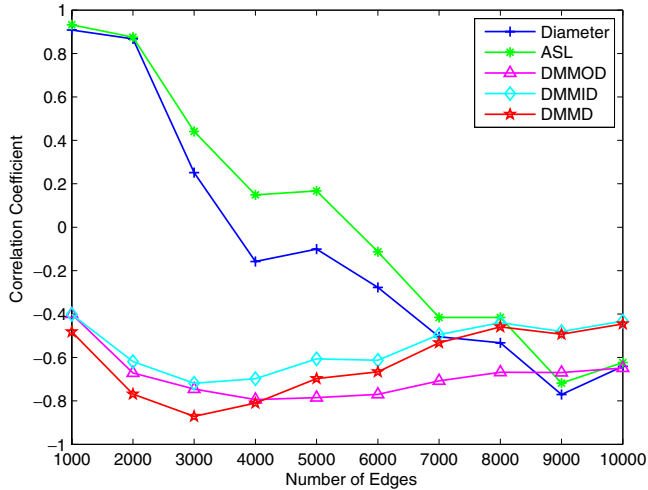


Fig. 4. Correlation coefficients of average propagation ratios with other statistical parameters of complex software networks. ASL denotes the average shortest length. DMMOD denotes the difference between the maximum and minimum out-degrees. DMMID denotes the difference between the maximum and minimum in-degrees. DMMD denotes the difference between the maximum and minimum degrees.

such correlation, where the correlation coefficient, a concept from statistics, is a measure of how well trends in the predicted values follow trends in past actual values. It is a measure of how well the predicted values from a forecast model “fit” with the real-life data.

All networks used in our numerical calculations have 1000 nodes. The following observations are

clearly evident from the results:

- (1) The correlation between the average propagation ratios and the network diameters (or average shortest lengths) gradually decreases with the number of the edges.
- (2) When the number of the edges is relatively small, the correlation is positive; however, when the number of the edges is relatively large, the correlation is negative.
- (3) The correlation between the average propagation ratios and the differences between the maximum and minimum out-degrees (or in- or both degrees) decreases initially and then gradually increases as the number of the edges further increases.

3.3. Relating average propagation ratios with network structures

In the foregoing, we have established some relationships between average propagation ratios and some conventional statistical parameters. It would be of interest to know how network structures affect average propagation ratios. Such knowledge will be useful for connecting statistical parameters with network structures, and hence is important for network design. In this subsection we will use some simplified structures to study the dependence of average propagation ratios on the kind of network structure. Figure 5 shows the average propagation

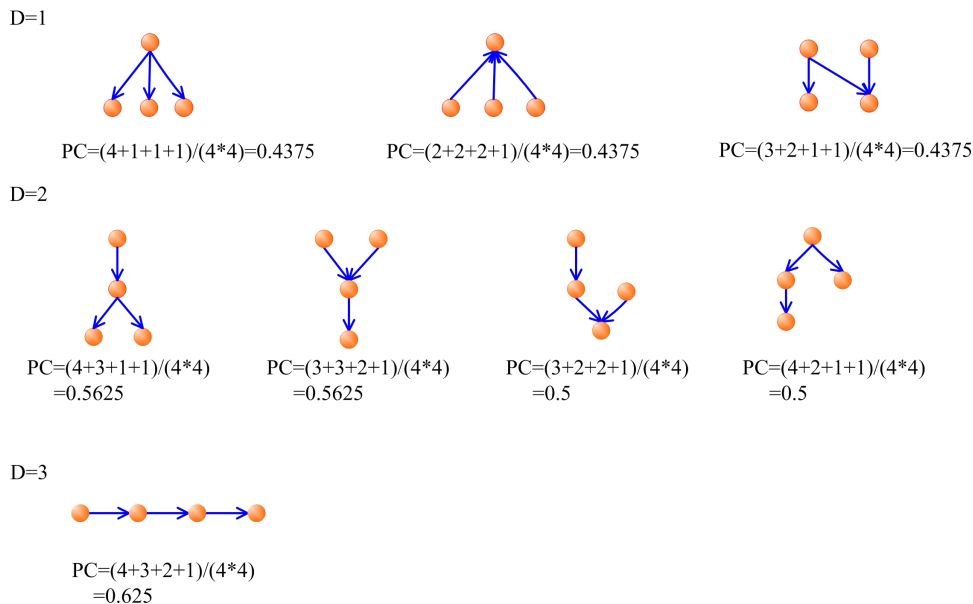


Fig. 5. Average propagation ratios of networks with four nodes and three edges, showing the fundamental difference between star and chain structures.

ratios of some networks with four nodes and three edges. Specifically, the average propagation ratios of the networks in Fig. 5 are 0.4375, 0.4375, 0.4375, 0.5625, 0.5625, 0.5, 0.5 and 0.625. Moreover, our results clearly point out that the star structure has much lower average propagation ratios and the chain structure has much higher average propagation ratios. These results remain generally valid for complex software networks.

4. Conclusions

In this paper we have introduced an efficient statistical measure, called *average propagation ratio*, to characterize the structural quality of general complex software networks. Several representative real-world complex software networks have been analyzed to illustrate the physical meaning of the proposed average propagation ratio. Also, we have further studied the relationships between the average propagation ratios and some statistical parameters. Our results reveal that (i) the correlation between the average propagation ratios and the network diameters (or average shortest lengths) gradually decreases with the increase of the number of the edges; (ii) the correlation between the average propagation ratios and the differences between the maximum and minimum out- (or in- or both) degrees initially decreases and then gradually increases as the number of the edges further increases. Moreover, we have shown that the star structure has much lower average propagation ratios and the chain structure has much higher average propagation ratios. These results provide useful guiding principles for enhancing the structural quality of complex software networks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 60304017, 20336040, 60221301, the National Key Basic Research and Development 973 Program of China under Grant 2006CB708302, the Scientific Research Startup Special Foundation on Excellent

PhD Thesis, and the Presidential Award of Chinese Academy of Sciences.

References

- Barabási, A. L. & Albert, R. [1999] “Emergence of scaling in random networks,” *Science* **286**, 509–512.
- Challet, D. & Lombardoni, A. [2004] “Bug propagation and debugging in asymmetric software structures,” *Phys. Rev. E* **70**, 46109.
- De Moura, A. P. S., Lai, Y. C. & Motter, A. E. [2005] “Signatures of small-world and scale-free properties in large computer programs,” *Phys. Rev. E* **68**, 17102.
- Liu, J., He, K. Q., Ma, Y. T. & Peng, R. [2006] “Scale free in software metrics,” *Proc. 30th Ann. Int. Comput. Software & Appl. Conf. (COMPSAC 2006)*, September 18–21, Chicago, USA, pp. 229–235.
- Lü, J., Yu, X., Chen, G. & Cheng, D. [2004] “Characterizing the synchronizability of small-world dynamical networks,” *IEEE Trans. Circuits Syst.-I* **51**, 787–796.
- Lü, J. & Chen, G. [2005] “A time-varying complex dynamical network model and its controlled synchronization criteria,” *IEEE Trans. Auto. Contr.* **50**, 841–846.
- MacCormack, A., Rusnak, J. & Baldwin, C. Y. [2006] “Exploring the structure of complex software designs: An empirical study of open source and proprietary code,” *Manag. Sci.* **52**, 1015–1030.
- Myers, C. R. [2003] “Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs,” *Phys. Rev. E* **68**, 46116.
- Potantin, A., Noble, J., Frean, M. & Biddle, R. [2005] “Scale-free geometry in OO programs,” *Commun. ACM* **48**, 99–103.
- Ulrich, K. [1995] “The role of product architecture in the manufacturing firm,” *Res. Pol.* **24**, 419–440.
- Valverde, S. & Solé, R. V. [2003] “Hierarchical small worlds in software architecture,” Arxiv preprint cond-mat/0307278.
- Valverde, S. & Solé, R. V. [2005] “Network motifs in computational graphs: A case study in software architecture,” *Phys. Rev. E* **72**, 26107.
- Watts, D. J. & Strogatz, S. H. [1998] “Collective dynamics of ‘small-world’ networks,” *Nature* **393**, 409–410.
- Zhou, J., Lu, J. & Lü, J. [2006] “Adaptive synchronization of an uncertain complex dynamical network,” *IEEE Trans. Auto. Contr.* **51**, 652–656.