Feedback arcs and node hierarchy in directed networks*

Jin-Hua Zhao(赵金华)[1,3] and Hai-Jun Zhou(周海军)[1,2,†]

[1] *Key Laboratory of Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing* 100190, *China*
[2] *School of Physical Sciences, University of Chinese Academy of Sciences, Beijing* 100049, *China*
[3] *Department of Applied Science and Technology, Politecnico di Torino,* 10129 *Torino, Italy*

(Received 11 May 2017; published online 27 May 2017)

Directed networks such as gene regulation networks and neural networks are connected by arcs (directed links). The nodes in a directed network are often strongly interwound by a huge number of directed cycles, which leads to complex information-processing dynamics in the network and makes it highly challenging to infer the intrinsic direction of information flow. In this theoretical paper, based on the principle of minimum-feedback, we explore the node hierarchy of directed networks and distinguish feedforward and feedback arcs. Nearly optimal node hierarchy solutions, which minimize the number of feedback arcs from lower-level nodes to higher-level nodes, are constructed by belief-propagation and simulated-annealing methods. For real-world networks, we quantify the extent of feedback scarcity by comparison with the ensemble of direction-randomized networks and identify the most important feedback arcs. Our methods are also useful for visualizing directed networks.

Keywords: directed graph, feedback arc, hierarchy, message-passing algorithm

PACS: 89.75.Fb, 75.10.Nr, 87.16.Yc, 02.70.Uu **DOI:** 10.1088/1674-1056/26/7/078901

1. Introduction

Directed networks are formed by nodes and arcs (i.e., directed links) pointing from one node to another. They are ubiquitous in biological and technological systems; for instance, neurons in the brain rely on directed synaptic connections to form an information-processing network,[1] and cell regulatory networks contain directed interactions between genes, proteins, and other small molecules.[2–4] Structural properties of directed networks at different scales have been studied in the literature for many years, especially on network small motifs,[4,5] mesoscopic communities,[6,7] strongly connected components,[8,9] and network hierarchical structures.[10–14] A directed network can easily be decomposed into a set of strongly connected components (SCCs). If one represents each SCC by a single vertex and neglects its internal connections, then at this coarse-grained level the directed network is simply a directed acyclic graph and there is no directed cycles among the SCCs.[9,13] Each SCC is itself a maximal subnetwork formed by some nodes and the arcs between them, and any node can reach and be reached by any other node of the same SCC through at least one directed path. Directed cycles are usually abundant in the large SCCs (each of which contains many nodes and arcs), and they cause strong feedback effect and make the information-processing dynamics in the network highly complex.[15,16]

The hierarchical structures within large SCCs of directed networks have not yet been fully investigated except for a few earlier efforts (e.g., Refs. [12], [14], and [16]–[18]). Due to the cyclic nature of a SCC, it appears at first sight to be quite ambiguous or even meaningless to order its nodes in a particular way and to define an intrinsic flow direction.[13] However in this paper we show that the arcs that are most vital for feedback interactions can be identified by collectively considering all the directed cycles of the network. We take an optimization approach based on the so-called principle of minimum feedback,[12] which defines the minimum feedback arc set problem. An integer hierarchical level is assigned to each node of the input network and the resulting level configuration of all the nodes is called a node hierarchy. The node levels in this hierarchy are optimized by two efficient physics-inspired algorithms, SA and BPD, which minimize the total number of feedback arcs (defined as those pointing from lower-level nodes to higher-level nodes).

Given a real-world directed network, we can construct many near-minimum feedback arc sets by repeatedly running the SA or BPD algorithm. The sizes of these constructed sets are very close to each other and are much smaller than the total number of arcs in the network. We also find that, while most of the arcs of the network never appear in any of these feedback arc sets, a few of them appear in almost all of them. As a concrete example, for the Florida food web[19] formed by 128 nodes and 2106 arcs, only six of the arcs need to be classified as feedbacks, (Fig. 1) which is much lower than the expected number of 601 feedback arcs in a direction-randomized network. Our algorithms reveal that two arcs of the food-web network are present in all the minimum feedback arc sets. Similar

*Project by the National Basic Research Program of China (Grant No. 2013CB932804) and the National Natural Science Foundations of China (Grant Nos. 11121403 and 11225526). J-H Zhao also acknowledges support by Fondazione CRT under project SIBYL, initiative "La Ricerca dei Talenti".

†Corresponding author. E-mail: zhouhj@itp.ac.cn

© 2017 Chinese Physical Society and IOP Publishing Ltd

results are obtained for other real-world networks.

By distinguishing feedforward arcs and feedback arcs for a real-world directed network, our work help to reveal the hidden principal direction of flows in the network and the hierarchical organization of the nodes within the strongly connected network components. For biological networks, the identified most important feedback arcs might serve as optimal targets of intervening the system.[20] Our algorithms are also useful for network visualization.[21] The source codes of these algorithms will be publicly available to facilitate analyzing and visualizing biological, technological, and social networks.

The minimum feedback principle serves as a useful framework for understanding the structural properties of directed networks. Since feedback interactions interfere with the default dynamics of a real-world complex system, it should be desirable to delete the redundant feedbacks to increase efficency and decrease cost. We therefore expect the minimum feedback principle to be evolutionarily plausible for real-world complex networks. Even if the feedback minimization is not a major driving force for network evolution and adaptation, the technical results of this work still hold.

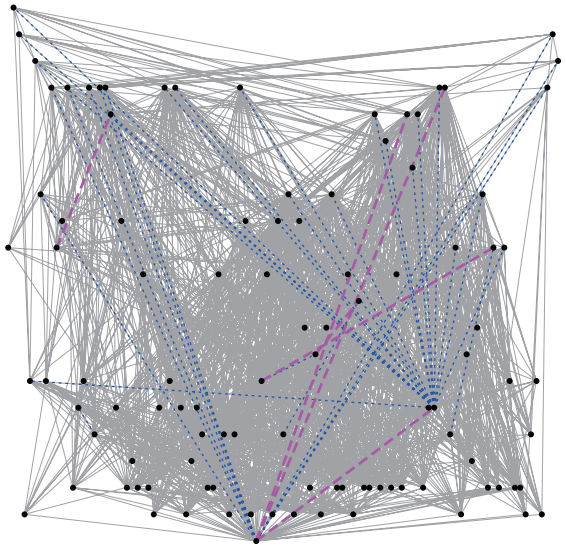


Fig. 1. (color online) An optimal node hierarchy for the Florida food web[19] and the corresponding feedback arcs. The whole network has 128 nodes and 2106 arcs, and its largest strongly connected component contains 103 nodes and 1579 arcs (for clarity only this component is shown). The nodes (black dots) are arranged to 22 hierarchical levels starting from level 0 at the bottom. Gray solid lines are feedforward arcs (pointing from higher-level nodes to lower-level nodes), and red dashed lines are feedback arcs (pointing from lower-level nodes to higher-level nodes). Each blue dotted line represents a pair of opposite arcs between two nodes.

2. Node hierarchy and feedback arcs

Given a directed network G of N nodes and M arcs, with arc density $\alpha \equiv M/N$, we introduce a node hierarchy $\underline{h} \equiv (h_1, h_2, \ldots, h_N)$ to partially order the N nodes $i \in \{1, 2, \ldots, N\}$. The level of each node i takes a nonnegative integer value $h_i \in \{0, 1, \ldots, N-1\}$, and level 0 is the lowest. A node j at positive level h_j must have at least one outgoing arc (j,k) to a node k at one level below (i.e., $h_k = h_j - 1$) to justify its level. Under these level constraints, our goal is to construct an optimal node hierarchy which agrees with most of the arcs, so that the total number of arcs from higher-level nodes to lower-level nodes reaches the global maximum value.

The node hierarchy problem is essentially equivalent to the feedback arc set problem, a fundamental and famous nondeterministic polynomial hard (NP-hard) problem in computer science[22] (Appendix A). A set Λ of arcs is regarded as a feedback arc set (FAS) if it intersects with every directed cycle of the network. Notice that if all the arcs of a FAS are deleted, the remaining network contains no directed cycle. A FAS is a minimal one if any of its proper subset is no longer a FAS; and it is a minimum one if its cardinality is the smallest among all the feedback arc sets. Given a node hierarchy $\underline{h}$ of network G, the set formed by all the arcs (i, j) with $h_i \leq h_j$ is a FAS. On the other hand, a unique node hierarchy can be constructed for any FAS by first deleting all the arcs of this set from the network and assigning the lowest level 0 to all the nodes which have no outgoing arc, followed by iteratively assigning the level $1, 2, \ldots$ to all the remaining nodes which have outgoing arcs only to nodes at lower levels. Indeed there is a one-to-one correspondence between node hierarchies and the so-called neat feedback arc sets (Appendix A). All the minimal and minimum feedback arc sets (and some special non-minimal ones) are neat, and therefore an optimal node hierarchy is equivalent to a minimum FAS.

3. Mean field theory and belief propagation

We can treat the node hierarchy problem as a statistical mechanical system. Let us define the energy of an arc (i, j) as $E_{ij}(h_i, h_j) = 0$ for $h_i > h_j$ and $E_{ij}(h_i, h_j) = 1$ for $h_i \leq h_j$. The total energy of hierarchy $\underline{h}$ is then the sum of arc energies

$$E(\underline{h}) \equiv \sum_{(i,j)\in G} E_{ij}(h_i, h_j). \tag{1}$$

We write down the following equilibrium partition function Z to combine the effects of energy and level constraints:

$$Z(\beta) = \sum_{h_1 \geq 0} \cdots \sum_{h_N \geq 0} \prod_{i=1}^{N} \psi_i \prod_{(j,k)\in G} \psi_{jk}. \tag{2}$$

Here ψ_i is the Boltzmann factor of node i due to its level constraint: $\psi_i = 1$ if $h_i = 0$ or i has an outgoing arc (i, j) to a node j at level $h_j = h_i - 1$; otherwise $\psi_i = 0$. The Boltzmann factor of arc (j,k) is $\psi_{jk} = 1$ if its energy is zero ($h_j > h_k$); otherwise $\psi_{jk} = \mathrm{e}^{-\beta}$ with the inverse temperature β being an adjustable parameter. Notice that each node hierarchy $\underline{h}$ contributes a weight $\mathrm{e}^{-\beta E(\underline{h})}$ to Z. At sufficiently large values of β, the node hierarchies with the global minimum energy value (i.e., the optimal node hierarchies) will have overwhelming contributions to the partition function Z.

We have solved model (2) by the replica-symmetric (RS) cavity method developed in the spin glass research field[23–28] (Appendix B). Due to the strong level constraints, the mean-field equations of this RS theory are very complicated and are computationally inefficient.

Fortunately, the equivalence between the optimal node hierarchy problem and the minimum FAS problem means that we can obtain a near-optimal node hierarchy by first constructing a near-minimum FAS. For the latter task, the level constraints of Eq. (2) are not necessary, so we can drop them by setting the Boltzmann factors of all the nodes i to be $\psi_i \equiv 1$. The RS mean-field theory for this relaxed model is much more convenient for numerical treatment (Appendix C). This simplified theory estimates the probability ρ_{ij} of arc (i,j) being a feedback arc to be

$$\rho_{ij} = \frac{\mathrm{e}^{-\beta} \sum_{h_i=0}^{D-1} \sum_{h_j=h_i}^{D-1} q_{i\to j}^{h_i} q_{j\to i}^{h_j}}{1-(1-\mathrm{e}^{-\beta}) \sum_{h_i=0}^{D-1} \sum_{h_j=h_i}^{D-1} q_{i\to j}^{h_i} q_{j\to i}^{h_j}}, \tag{3}$$

where the integer D restricts the level of each node i to be $h_i < D$ to compensate for the removed level constraints; the function $q_{j\to j'}^{h_j}$ denotes the probability that node j will be at level h_j if node j' is absent. The self-consistent belief propagation (BP) equation for this cavity probability is

$$q_{j\to j'}^{h_j} \propto \prod_{i\in p(j)\backslash j'} \left[1-(1-\mathrm{e}^{-\beta}) \sum_{h_i=0}^{h_j} q_{i\to j}^{h_i}\right] \times \prod_{k\in c(j)\backslash j'} \left[1-(1-\mathrm{e}^{-\beta}) \sum_{h_k=h_j}^{D-1} q_{k\to j}^{h_k}\right], \tag{4}$$

where $p(j) \equiv \{i : (i,j) \in G\}$ and $c(j) \equiv \{k : (j,k) \in G\}$; and $p(j)\backslash j'$ is the subset of $p(j)$ with j' being excluded, similarly for $c(j)\backslash j'$.

We can iterate the BP equation (4) on the network G at a fixed large value of D (e.g., $D = 200$) and different values of β. The mean fraction ρ of feedback arcs at inverse temperature β is estimated by

$$\rho = \frac{1}{M} \sum_{(i,j)\in G} \rho_{ij}. \tag{5}$$

Based on Eqs. (3) and (4), a belief-propagation–guided decimation (BPD) algorithm is also implemented to construct near-minimum feedback sets. Briefly speaking, at each decimation step a tiny fraction of arcs (i,j) with the largest estimated ρ_{ij} values are deleted from the network G; then G is further simplified by deleting all the nodes which have no incoming or outgoing arc; then equation (4) is iterated a small number of times and the value of ρ_{kl} for each remaining arc (k,l) is updated. Details of the BPD algorithm can be found in Appendix D.

4. Simulated annealing

Let us represent an N-node permutation as a column vector $\mathcal{P} \equiv (v_1, v_2, \ldots, v_N)^{\mathrm{T}}$ with $v_r \in \{1,2,\ldots,N\}$ and $v_r \neq v_{r'}$ if $r \neq r'$. Another way of simplifying the level constraints of Eq. (2) is to set the level h_i of each node i to be its vertical position in $\mathcal{P}$. A most convenient way of permutating the nodes to reduce the total arc energy is simulated annealing (SA).[29] This method has been successfully applied on the directed and undirected feedback vertex set problems.[28,30,31] For the present FAS problem we follow the simple recipe of Ref. [30] (Appendix E). Starting from an initial random permutation and an initial low inverse temperature β, at each time step two rejection-free updating processes are performed: 1) an upward arc (i,j) with $h_i < h_j$ is chosen among all such arcs with probability proportional to $\exp\left[-\beta \max(0, s_{(i,j)}^{i\uparrow})\right]$ and node i is moved to be immediately above node j in permutation $\mathcal{P}$, where $s_{(i,j)}^{i\uparrow}$ is the increase in the number of upward arcs caused by this move; and 2) an upward arc (i',j') is chosen among all such arcs with probability proportional to $\exp\left[-\beta \max(0, s_{i',j'}^{j'\downarrow})\right]$ and j' is moved to be immediately below node i' in $\mathcal{P}$, where again $s_{(i',j')}^{j'\downarrow}$ is the increase in the number of upward arcs caused by this move. After $c_0 N$ such time steps (e.g., $c_0 = 5$ or even larger) the inverse temperature is increased to $\beta \leftarrow \beta/c_1$ (e.g., $c_1 = 0.99$). The search process terminates at a sufficiently large value of β.

5. Results on random network instances

We first test the BPD and SA algorithms on directed Erdös–Rényi (ER), directed regular random (RR) and directed scale-free (SF) random networks.[32,33] Both ER and RR networks are homogenous, while SF networks are quite heterogeneous in that some nodes have a lot of attached arcs (Appendix F). As the arc directions are completely random, no intrinsic flow direction should exist in these artificial networks. Our motivation here is to check whether near-minimum feedback arc sets can be achieved by BPD and SA.

We find that BPD and SA perform almost equally good on all the heterogeneous (SF) random networks and on the homogeneous (ER and RR) networks of arc density $\alpha < 5$; the fractions ρ of feedback arcs in the constructed FAS solutions are very close to the predicted values by the RS mean-field theory, indicating that nearly optimal solutions are indeed achieved (Fig. 2). BPD and SA greatly outperform the local degree-based heuristic (DH) which recursively deletes the arc (i,j) with the highest value of $d_i^{\mathrm{in}} \times d_j^{\mathrm{out}}$ from the network to destroy all directed cycles,[34] with the in-degree d_i^{in} and out-degree d_i^{out} being, respectively, the number of incoming and outgoing arcs of node i.

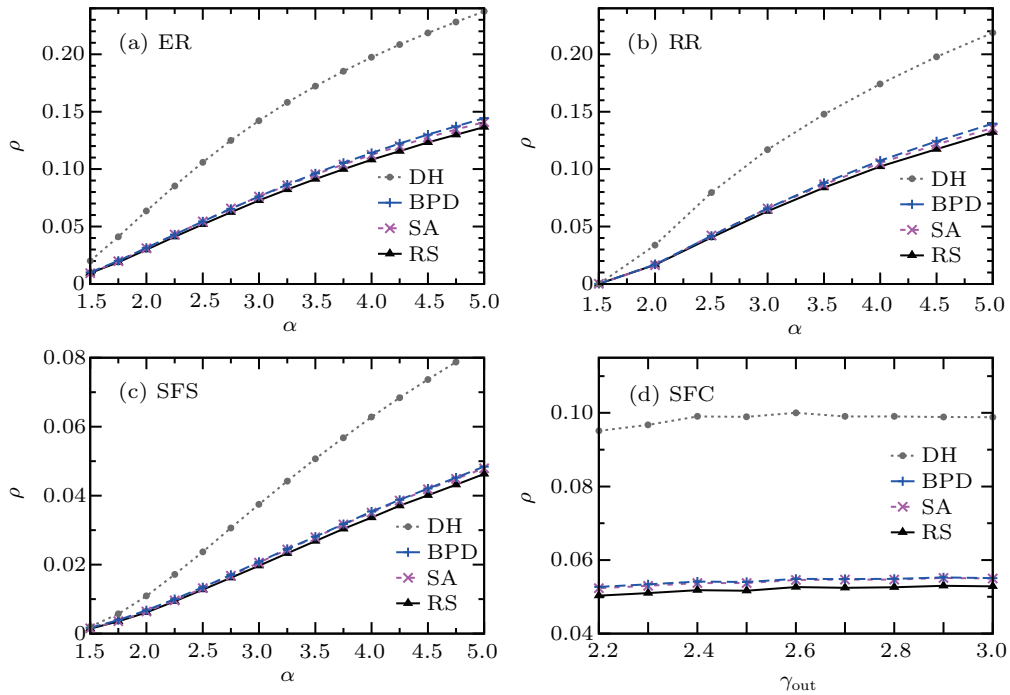


Fig. 2. (color online) Numerical results for random directed networks. α, arc density; ρ, fraction of feedback arcs. Algorithmic results of the local DH (circles), BPD (pluses), and SA (crosses) are compared with the predictions of the RS mean-field theory (triangles). Level upper-bound $D = 200$ and inverse temperature $\beta \approx 50$ for the BPD algorithm and the RS theory. Each data point is the average over 40 network instances of size $N = 10^4$; standard deviation (not shown) is less than 4×10^{-3}. Four ensembles of random networks are considered: (a) Erdös–Rényi (ER); (b) regular random (RR); (c) scale-free static (SFS[33]) with in- and out-degree exponents $\gamma_{\text{in}} = 2.5$ and $\gamma_{\text{out}} = 3.0$; and (d) scale-free configurational (SFC[32]) with in-degree exponent $\gamma_{\text{in}} = 2.5$ and different out-degree exponents γ_{out} and minimum in- and out-degree $d_{\text{min}} = 2$ and in- and out-degree upper-bound $d_{\text{max}} = \sqrt{N}$.

The SA algorithm slightly outperforms BPD for directed ER and RR networks of arc density $\alpha \geq 5$. For ER networks of arc density $\alpha = 5.0$, the typical fraction ρ of feedback arcs in solutions constructed by SA has the value $\rho \approx 0.1409$, while the corresponding value for the BPD-obtained solutions is $\rho \approx 0.1445$. We can improve the performance of BPD to a small extent by choosing a larger value D of level upper-bound, but the computation cost increases linearly with D. To further boost the performance of the BPD algorithm and beat the SA algorithm, we need to design a better statistical physics model for the feedback arc set problem. We plan to explore this challenging issue in a future paper.

In terms of computation time, the BPD and SA algorithms both are efficient on relatively small random directed network instances. They usually reach satisfactory solutions within one or two hours when applied on networks of size $N = 10^4$ vertices and arc densities $\alpha \leq 10$. For large network instances, however, the BPD algorithm is much faster than the SA algorithm. For example, given a RR network of size $N = 10^5$ and arc density $\alpha = 5$, the SA algorithm needs a computing time of 53.2 hours to construct a FAS solution while the BPD algorithm only needs a much shorter time of 21.5 hours (the test was performed on a desktop computer with Intel-6300 1.86 GHz CPU and 2 GB memory).

6. Results on real-world network instances

As a demonstration of practical applications, we now apply BPD and SA on a small set of representative real-world directed networks (Table 1).

Regulatory: the epidermal growth factor receptor (EGFR) signal transduction network,[3,15] with $N = 61$ nodes and $M = 112$ arcs. Each node represents a molecular species such as kinases, phosphatase, and ions; each arc represents a directed regulatory interaction between two molecular species.

Table 1. Solving the node hierarchy problem for real-world networks. N, node number; M, arc number; M_{s}, number of simple arcs (which have no opposite counterpart); $|\Lambda|$, number of simple feedback arcs; $|\Lambda^{\text{Rd}}|$ and σ^{Rd}, expected number of simple feedback arcs in a direction-randomized network and its standard deviation; R, scarcity extent of feedback arcs. Simulation results are all obtained by the SA algorithm.

Network	N	M	M_{s}	$\lvert\Lambda\rvert$	$\lvert\Lambda^{\text{Rd}}\rvert$	σ^{Rd}	R
Regulatory	61	112	108	7	5.4	1.6	−1.0
Food web	128	2106	2044	6	601	9	63
Neural	297	2359	1951	70	405	10	34
Circuit	512	819	819	32	24.9	3.9	−1.8
Metabolic	1469	3447	3383	555	315	10	−24
Wiki-Vote	7115	103689	97835	3040	32185	74	392
P2P-share	62586	147892	147892	2269	13820	67	172

Food web: the Florida Bay ecosystem network,[19] containing $N = 128$ nodes and $M = 2106$ arcs. Each node represents a species (such as bacteria, zooplankton, shrimp) or a molecular type such as particular organic carbon, and each di-

rected arc represents transfer of biomass between two kinds of species or molecules.

Neural: the neural network of the nematode *C. elegans*,[5] containing $N = 297$ nodes and $M = 2359$ arcs. Each node represents a neural cell and each arc represents a directed connection between two neurons.

Circuit: the electronic sequential logic circuit network EC-s838,[5] containing $N = 512$ nodes and $M = 819$ directed connections.

Metabolic: the metabolic network of the nematode *C. elegans*,[10] with $N = 1469$ nodes and $M = 3447$ arcs. Each node represents a chemical molecule or an enzyme, and each arc means that a given molecule participates in a particular enzyme-catalyzed reaction or is produced by this reaction.

Wiki-Vote: the network of who-votes-on-whom among the Wikipedia administrators,[35] containing $N = 7115$ nodes and $M = 103689$ arcs.

P2P-share: the Gnutella peer-to-peer file sharing network,[36] containing $N = 62586$ nodes and $M = 147892$ arcs. Each node represents a computer server and each arc represents directed file transfer between two servers.

For these real-world network instances, we again find that the feedback arc sets contructed by BPD and SA are of very similar sizes. It is very likely that near-optimal FAS solutions have been achieved by these two algorithms. The SA algorithm and BPD perform equally good on the four small network instances, but SA slightly outperforms BPD on the three large network instances (Metabolic, Wiki-Vote, P2P-share). We list in Table 1 the results obtained by a single running of the SA algorithm on the examined real networks, where M_s and $|\Lambda|$ respectively denote the total number of simple arcs and simple feedback arcs (excluding all the bi-directional arcs).

For each examined real-world network we also generate 96 replicas with the same connectivity pattern but completely randomized directions of all the simple arcs, and apply SA on them to obtain the expected number $|\Lambda^{\mathrm{Rd}}|$ of simple feedback arcs and its standard deviation σ^{Rd}. The scarcity R of feedback arcs in the original network is then quantified as

$$R = \frac{|\Lambda^{\mathrm{Rd}}| - |\Lambda|}{\sigma^{\mathrm{Rd}}}. \tag{6}$$

This quantity has a clear statistical meaning. A large positive value of R suggests that the number $|\Lambda|$ of feedback arcs in the original network is significantly lower than the expected number $|\Lambda^{\mathrm{Rd}}|$ of feedback arcs in a direction-randomized network. Similarly, a highly negative R value suggests that feedback arcs are significantly more abundant in the original network than in a direction-randomized network.

As Table 1 reveals, feedback arcs are very rare in the Florida food web,[19] the *C. elegans* neural network,[5] and social networks Wiki-Vote[35] and P2P-share,[36] which all have very large positive R values. Reducing the number of feedback connections might enhance the efficiency of information processing in neural and social networks. On the other hand, feedback arcs are strongly enriched in the *C. elegans* metabolic network,[10] which has a highly negative R value. It may be necessary to have an abundant number of feedback connections to finely regulate the concentrations of cellular molecules.

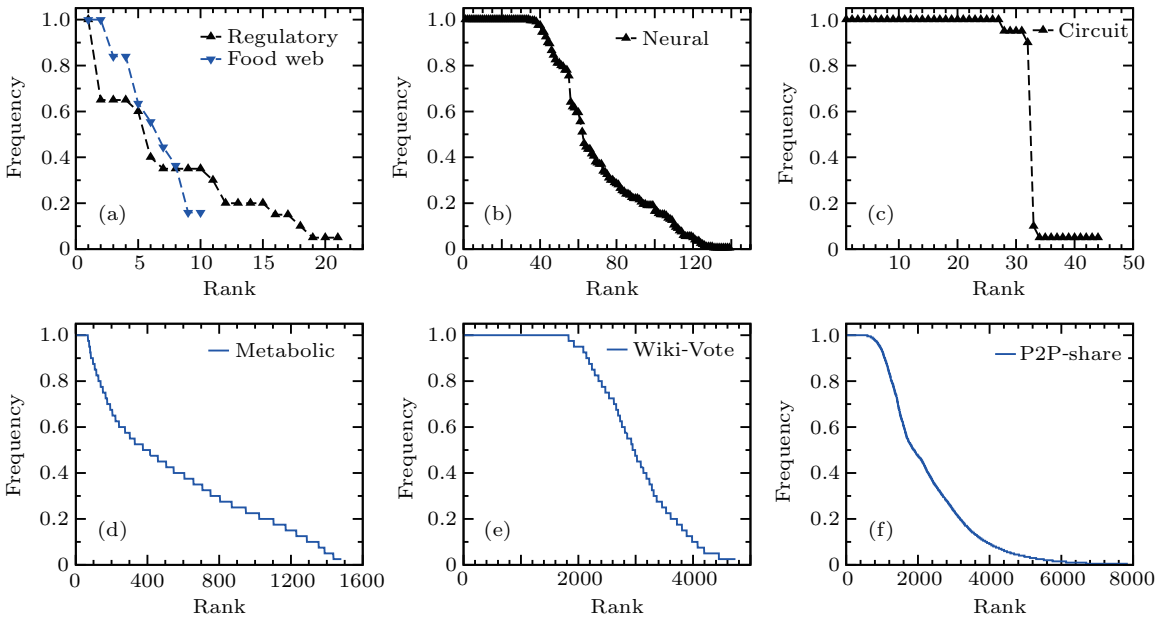


Fig. 3. (color online) Rank plot on the frequency (probability) ρ_{ij} of each arc (i, j) being a feedback arc. The arcs are ranked in decreasing order according to its frequency. The results are obtained by running the SA algorithm independently for 200 times on the same input network instance. (a) Regulatory network (FAS cardinality: 7) and Food web network (FAS cardinality: 6). (b) Neural network (FAS cardinality, mean and standard deviation: 70.0 ± 0.4). (c) Circuit network (FAS cardinality: 32 ± 1). (d) Metabolic network (FAS cardinality: 556 ± 2). (e) Wiki-Vote network (FAS cardinality: 3038 ± 2). (f) P2P-share network (FAS cardinality: 2266 ± 6).

When we repeatedly run the SA or BPD algorithm on the same real-world network instance, we find that the output feedback arc sets are usually not identical although their sizes are almost equal. Most importantly, we find that most of the arcs in the network never appear in any of these constructed feedback arc sets, but some arcs are present in almost all these sets (Fig. 3). These results strongly indicate that the arcs in a real-world network have very different significance in terms of the feedback role, and our SA and BPD algorithms can identify a small set of most important feedback arcs.

Figure 3 indicates that the near-optimal FAS solutions obtained by SA or BPD are considerably degenerate. We expect the true optimal FAS solutions will also be degerate. An important reason for this degeneracy is the presence of many long directed strings (non-branched paths) formed by vertices whose in- and out-degree both are unity. If one part of a directed cycle is such a non-branched path, there are many degenerate ways of breaking this cycle. Notice that the directed non-branched paths do not affect the overall node hierarchy of the network nor the principal flow direction of the network. We can reduce this degeneracy by replacing each directed non-branched path from one node i to another node j by a single coarse-grained arc pointing from i to j.

After the feedback probability ρ_{ij} for every arc (i,j) of a real-world network has been computed (through repeatedly running SA or BPD or, more efficiently, through employing Eq. (3) and BP iteration), the feedforward part (the backbone[12]) of the network can easily be constructed by checking every arc of the network in increasing order of the feedback probability and adding it to the backbone if no directed cycle will be formed.

7. Conclusion and outlook

In this paper, we introduced the optimal node hierarchy problem, which is essentially equivalent to the minimum feedback arc set problem, and presented two physics-based algorithms to efficiently solve this problem for random and real-world directed networks. Our BPD and SA algorithms are capable of revealing the hidden hierarchical structure and the principal flow direction of a real-world directed network. Our methods can also be used to discover a small number of arcs which are involved most significantly in feedback interactions. We found that feedback interactions are extremely supressed in some real-world networks.

The methods of this work may have wide practical applications in studies of biological, technological, and social networks and in network engineering. For example, after the intrinsic flow direction in the network has been determined, it may become much more easier to design efficient arc-deletion or arc-addition strategies to improve the functionality of the network and to make it more robust against random failures or intentional attacks. The key feedback arcs identified by our algorithms may serve as optimal targets of intervening the dynamical processes on the network.

A natural extension of the present work is to consider optimal ways of cutting long directed cycles to dismantle a directed network. Similar to the proposal of optimally dismantling an undirected network,[37,38] we may iteratively delete the arcs that are predicted to be most important for long-range feedback interactions to break the original directed network down into many small strongly connected components. Detailed numerical study on this important network optimization problem will be reported in a separate paper.

Directed cycles are large-scale structural aspects of a directed network. They cause complicated global constraints to the node hierarchy and FAS problems. Further efforts are needed to improve the theoretical models and the BPD algorithm of this paper. Indeed the two spin glass models of the present paper still have major shortcomings. Firstly, each node i of the network can take many different level states h_i, which considerably slows down the numerical computation. Secondly, the predicted minimum cardinalities of feedback arc sets by the two models differ noticably with each other and with the algorithmic results of BPD and SA. Thirdly, the associated BPD algorithms of the two models perform worse than the SA algorithm on homogeneous random networks of relative large arc densities. We hope these issues will be overcome in the near future by a refined statistical physics model of the minimum feedback arc set problem.

Appendices

We describe in the following appendices the technical details of the replica-symmetric (RS) mean-field theories, the belief-propagation-guided decimation (BPD) and the simulated annealing (SA) algorithms. Some technical details on generating random directed network instances and on visualizing directed networks are also given. The source codes of the BPD and SA algorithms will be made publicly available at http://power.itp.ac.cn/∼zhouhj/codes.html.

A directed network G is composed of N nodes and M arcs, each of which is a directed link pointing from one node (say i) to another node (say j) and is denoted as (i,j). The arc density is denoted as $\alpha \equiv M/N$. Given an arc (i,j), we say that node j is a downstream neighbor (child) of i and node i an upstream neighbor (parent) of j. Let us denote by $c(j) \equiv \{k : (j,k) \in G\}$ the set of downstream neighbors (childrens) of node j, and similarly denote by $p(j) \equiv \{i : (i,j) \in G\}$ the set of upstream neighbors (parents) of node j. The in-degree d_j^{in} of node j is then the cardinality of set $p(j)$, that is $d_j^{\text{in}} \equiv |p(j)|$, and the out-degree of node j is $d_j^{\text{out}} \equiv |c(j)|$.

An arc (i,j) from node i to node j is referred to as a simple arc if (and only if) the oppositely directed arc (j,i) from

node j to node i is absent. If both (i,j) and (j,i) are present in the network, a trivial directed cycle involving i and j will be formed, and one of these two arcs will be a feedforward arc and the other one will be a feedback arc. Because of this reason, in the present work we only consider simple arcs and neglect all the pairs of oppositely directed arcs.

Appendix A: Relationship between the node hierarchy problem and the feedback arc set problem

A node hierarchy $\underline{h} = (h_1, h_2, \ldots, h_N)$ is a hierarchy level configuration involving all the nodes of network G. A node hierarchy $\underline{h}$ must satisfy the following two sets of constraints: 1) the level of each node i must be a non-negative integer, namely, $h_i \in \{0,1,2,\ldots,N\}$; 2) a node j at positive level $h_j \geq 1$ must have outgoing arc(s) to node(s) at one level below (in other words, there must be at least one arc (j,k) from j to a node k at level $h_k = h_j - 1$).

A feedback arc set (FAS) is a set Λ of arcs with the property that if all the arcs in Λ are deleted from network G, the remaining subnetwork will be free of any directed cycle. Two feedback arc sets for a small network are shown in Fig. A1 as examples. A minimal FAS is a FAS Λ of minimal cardinality, meaning that any proper subset of Λ is no longer a FAS. A minimum FAS is a special minimal FAS whose cardinality is the smallest among all the feedback arc sets.

The concept of node hierarchy is closely related to the concept of feedback arc set. Given a FAS Λ, a unique node hierarchy $\underline{h}$ can be constructed through the following iteration process:

a. All the arcs of set Λ are deleted from network G, resulting in a directed acyclic subnetwork G'.

b. Some of the nodes must have no outgoing arc in subnetwork G', and they are all assigned the lowest hierarchy level 0. Then the level parameter h is set to be $h = 1$.

c. If some nodes were not assigned a level during the preceding step(s), then some of these remaining nodes must only have outgoing arcs in G' to the assigned nodes (whose levels are surely less than h), and all such nodes are assigned the hierarchy level h. Then h is increased by one ($h \leftarrow h+1$).

d. Return to step c as long as some nodes are still not yet assigned a hierarchy level.

The resulting hierarchy level configuration $\underline{h} = (h_1, h_2, \ldots, h_N)$ of the N nodes must be a node hierarchy, as each node i has integer level $h_i \geq 0$ and each node j of positive level h_j has at least one outgoing arc (j,k) to a node k of level $h_k = h_j - 1$. We have therefore proved that every FAS can be mapped to a unique node hierarchy. Let us denote by $\underline{h}(\Lambda)$ the mapped node hierarchy of the FAS Λ. Because of the one-to-one mapping from Λ to $\underline{h}(\Lambda)$, the node hierarchy $\underline{h}(\Lambda)$ contains all the information of Λ.

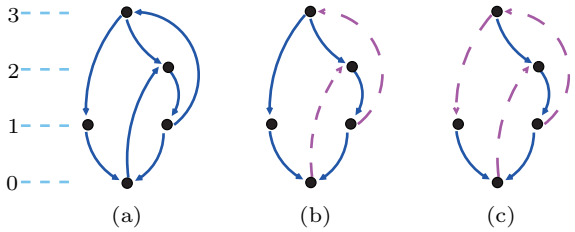


Fig. A1. (color online) One-to-many mapping between node hierarchies and feedback arc sets. (a) A node hierarchy for a small directed network with $N = 5$ nodes and $M = 7$ arcs: the number of nodes at level 0, 2, and 3 is one, while the number of nodes at level 1 is two. (b) The two dashed arcs form a feedback arc set. (c) The three dashed arcs form another feedback arc set.

In this work we regard a feedback arc set Λ as neat if, and only if, for each arc $(i,j) \in \Lambda$ the level of the upstream node i does not exceed that of the downstream node j (namely, $h_i \leq h_j$) in the uniquely determined node hierarchy $\underline{h}(\Lambda)$. Notice that a neat FAS of network G is not necessarily a minimal FAS of G. (For example, the arc set containing all the arcs of the network is a neat FAS, but it is not a minimal FAS.) On the other hand, every minimal FAS must be a neat FAS. As a corollary, every minimum FAS is a neat FAS. The concept of neat FAS can be understood as a natural extension of the concept of minimal FAS.

We now prove that there is a one-to-one correspondence between a neat FAS Λ and a node hierarchy $\underline{h}$, which means that the "function" $\underline{h} = \underline{h}(\Lambda)$ is invertible if Λ is restricted to be a neat FAS.

First, let us emphasize that, although there is a one-to-one mapping from FAS to node hierarchy, the mapping from node hierarchy to FAS is not one-to-one but instead many-to-one. In other words, many different feedback arc sets are compatible with the same node hierarchy $\underline{h}$. Figure A1 gives a clear demonstration of this important fact. However, given a node hierarchy, say $\underline{h}^{(1)} = (h_1^{(1)}, h_2^{(1)}, \ldots, h_N^{(1)})$, a unique and neat FAS (denoted as $\Lambda^{(1)}$) can be constructed through the following simple process, starting from $\Lambda^{(1)} = \emptyset$: For each arc $(i,j) \in G$ we add it to $\Lambda^{(1)}$ if and only if $h_i^{(1)} \leq h_j^{(1)}$. It is straightforward to check that the resulting set $\Lambda^{(1)}$ must be unique, it must be a FAS, and it must be neat. If one applies on $\Lambda^{(1)}$ the above-mentioned mapping of FAS to node hierarchy, the resulting node hierarchy $\underline{h}^{(2)} \equiv \underline{h}(\Lambda^{(1)})$ will be identical to $\underline{h}^{(1)}$. This last statement can be verified by the following iterative reasoning:

1. If a node i has level $h_i^{(1)} = 0$ in node hierarchy $\underline{h}^{(1)}$, then all its outgoing arcs must belong to the neat FAS $\Lambda^{(1)}$ and therefore it will be assigned the hierarchy level $h_i^{(2)} = 0$ when $\Lambda^{(1)}$ is mapped back to a node hierarchy. Therefore $\underline{h}(\Lambda^{(1)})$ is identical to $\underline{h}^{(1)}$ at hierarchy level 0.

2. If a node j has level $h_j^{(1)} = 1$ in node hierarchy $\underline{h}^{(1)}$, then at least one of its outgoing arc, say (j,k), points to a node k of level $h_k^{(1)} = 0$ and therefore does not belong to $\Lambda^{(1)}$; on the other hand, if node j has an outgoing arc, say (j,l), to a

node l of level $h_l^{(1)} \geq 1$, this arc must belong to $\Lambda^{(1)}$. Because of these two properties, node j will be assigned the hierarchy level $h_j^{(2)} = 1$ in the mapping $\underline{h}(\Lambda^{(1)})$. This means that $\underline{h}(\Lambda^{(1)})$ is identical to $\underline{h}^{(1)}$ also at hierarchy level 1.

3. The reasoning of the preceding step (2) can be applied to nodes at the hierarchy level $h = 2,3,\ldots$ of $\underline{h}^{(1)}$ to confirm that $\underline{h}(\Lambda^{(1)})$ is identical to $\underline{h}^{(1)}$ at all these hierarchy levels.

We have therefore completed the proof of one-to-one correspondence between a node hierarchy $\underline{h}$ and a neat FAS Λ. In this work we regard a node hierarchy $\underline{h}$ as minimal if, and only if, its corresponding neat FAS Λ is a minimal FAS. A node hierarchy $\underline{h}$ is referred to as an optimal (or minimum) node hierarchy if and only if the corresponding neat FAS Λ is a minimum FAS.

Since every minimal FAS is a neat FAS, the nice property of one-to-one correspondence between node hierarchy and neat FAS means that, the problem of constructing an optimal (or nearly optimal) node hierarchy is essentially equivalent to the problem of constructing a minimum (or nearly minimum) FAS. Because the FAS problem is a NP-complete combinatorial optimization problem, the node hierarchy problem must also be NP-complete.

Appendix B: Replica-symmetric mean field theory for the strongly constrained model

Let us refer to the strongly constrained statistical physics system (2) as model-R (the "restricted" model). According to this model, each node j of the directed network G has a hierarchical level $h_j \geq 0$, and the level constraint associated with node j is: if j is at a positive level ($h_j > 0$) then it must have at least one outgoing arc to a node at one level below (i.e., it must have an outgoing arc (j,k) to a node k at level $h_k = h_j - 1$). The partition function of model-R at inverse temperature β is

$$Z_{\text{modelR}}(\beta) = \sum_{h_1=0}^{D-1} \cdots \sum_{h_N=0}^{D-1} \prod_{(k,l)\in G} \psi_{kl}(h_k,h_l) \times \prod_{i=1}^{N} \Big[1 - (1-\delta_0^{h_i}) \prod_{j\in c(i)} (1-\delta_{h_j+1}^{h_i})\Big], \quad \text{(B1)}$$

where the Kronecker symbol $\delta_m^n = 1$ for $m = n$ and $\delta_m^n = 0$ for $m \neq n$; the Boltzmann factor $\psi_{kl}(h_k,h_l) = 1$ for $h_k > h_l$ and $\psi_{kl}(h_k,h_l) = \mathrm{e}^{-\beta}$ for $h_k \leq h_l$. Equation (B1) is identical to Eq. (2) of the main text. The integer adjustable parameter D is a level upper-bound introduced for computational convenience. In principle we should set the level upper-bound D to be $D = N$; but it turns out that the theoretical results are not sensitive to the precise value of D (see Fig. B2), and actually too large a value of D will deteriorate the performance of the associated BPD message-passing algorithm (see Fig. B3).

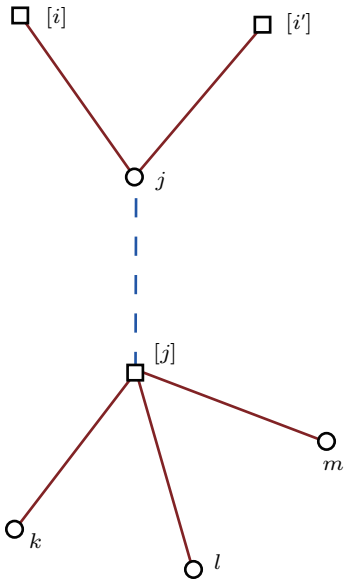


Fig. B1. (color online) A bipartite-graph representation of the strongly constrained model (B1), showing the neighborhood structure of a node j. In the original directed network G, node j has three downstream neighbors ($d_j^{\text{out}} = 3$ and $c(j) = \{k,l,m\}$) and two upstream neighbors ($d_j^{\text{in}} = 2$ and $p(j) = \{i,i'\}$). The circles represent node j and its three downstream nodes k, l, and m. The squares represent the level constraints $[j],[i],[i']$ associated with nodes j and all its upstream neighbors. The square for constraint $[j]$ is connected to node j by a dashed line and to the nodes in set $c(j)$ by solid lines.

In the following discussions, let us denote by $[j]$ the level constraint associated with node j. This constraint induces many-body interactions among j and all its downstream neighbors in the set $c(j)$. We can represent model (B1) by a bipartite graph involving constraint nodes (squares) and variable nodes (circles) and the edges between the squares and circles, see Fig. B1.

B1. The belief-propagation equation

Let us denote by $p_{[j]\to j}^{h_j}$ the probability that node j will be at level h_j if it is only constrained by the constraint $[j]$. Similarly, for each downstream neighbor k of node j, we denote by $p_{[j]\to k}^{h_k}$ the probability that k will be at level h_k if it is only constrained by the constraint $[j]$. If node j has no downstream neighbor (i.e., its out-degree $d_j^{\text{out}} = 0$), then $p_{[j]\to j}^{h_j} = \delta_0^{h_j}$. In the general case of $d_j^{\text{out}} \geq 1$, if we assume that all the nodes attached to the constraint $[j]$ (see Fig. B1) are mutually independent in the absence of this constraint, we can write down the following set of belief propagation (BP) equations[23–28]

$$p_{[j]\to j}^{h_j} = \frac{1}{z_{[j]\to j}} \sum_{\{h_k : k\in c(j)\}} \Big[1 - (1-\delta_0^{h_j}) \prod_{k\in c(j)} (1-\delta_{h_k+1}^{h_j})\Big] \times \prod_{k\in c(j)} \Big[\mathrm{e}^{-\beta E_{jk}} q_{k\to[j]}^{h_k}\Big], \quad \text{(B2a)}$$

$$p_{[j]\to k}^{h_k} = \frac{1}{z_{[j]\to k}} \sum_{h_j} \mathrm{e}^{-\beta E_{jk}} q_{j\to[j]}^{h_j} \times \sum_{\{h_{k'} : k'\in c(j)\setminus\{k\}\}} \Big[1 - (1-\delta_0^{h_j}) \prod_{k'\in c(j)} (1-\delta_{h_{k'}+1}^{h_j})\Big]$$

$$\times \prod_{k' \in c(j)\backslash k} \left[\mathrm{e}^{-\beta E_{jk'}} q_{k' \to [j]}^{h_{k'}} \right], \tag{B2b}$$

where $z_{[k]\to k}$ and $z_{[k]\to j}$ are two probability normalization constants; $c(j)\backslash k$ means the subset of $c(j)$ with node k being excluded; E_{jk} is the energy of arc (j,k) which is $E_{jk}=0$ for $h_j > h_k$ and $E_{jk}=1$ for $h_j \le h_k$; and

$$q_{j\to[j]}^{h_j} \equiv \prod_{i\in p(j)} p_{[i]\to j}^{h_j}, \quad q_{k\to[j]}^{h_k} \equiv p_{[k]\to k}^{h_k} \prod_{j' \in p(k)\backslash j} p_{[j']\to k}^{h_k}, \tag{B3}$$

with $p(k)\backslash j$ being the subset of $p(k)$ with node j being excluded. The quantity $q_{j\to[j]}^{h_j}$ actually is proportional to the probability that node j will be at level h_j if it is not constrained by the constraint $[j]$; similarly, $q_{k\to[j]}^{h_k}$ is proportional to the probability that node k will be at level h_k if it is not constrained by the constraint $[j]$.

The BP equation (B2) can be rewritten in the following equivalent form which is more convenient for numerical implementation:

$$p_{[j]\to j}^{h_j} \propto \prod_{k\in c(i)} \left[\sum_{h_k} \mathrm{e}^{-\beta E_{jk}} q_{k\to[j]}^{h_k} \right] - (1-\delta_0^{h_j}) \times \prod_{k\in c(j)} \left[\sum_{h_k} \mathrm{e}^{-\beta E_{jk}} q_{k\to[j]}^{h_k} - q_{k\to[j]}^{h_j-1} \right], \tag{B4a}$$

$$p_{[j]\to k}^{h_k} \propto \sum_{h_j} \mathrm{e}^{-\beta E_{jk}} q_{j\to[j]}^{h_j} \left\{ \prod_{k'\in c(j)\backslash k} \left[\sum_{h_{k'}} \mathrm{e}^{-\beta E_{jk'}} q_{k'\to[j]}^{h_{k'}} \right] - (1-\delta_0^{h_j})(1-\delta_{h_k+1}^{h_j}) \times \prod_{k'\in c(j)\backslash k} \left[\sum_{h_{k'}} \mathrm{e}^{-\beta E_{jk'}} q_{k'\to[j]}^{h_{k'}} - q_{k'\to[j]}^{h_j-1} \right] \right\}. \tag{B4b}$$

B2. Thermodynamic quantities

The probability ρ_{jk} of arc (j,k) being a feedback arc is equal to the probability of $h_j \le h_k$. According to the RS mean-field theory (i.e., assuming that all the attached nodes of the constraint $[j]$ in Fig. B1 are mutually independent in the absence of this constraint), we have

$$\rho_{jk} = \frac{\mathrm{e}^{-\beta}}{z_j} \sum_{h_j\ge 0} q_{j\to[j]}^{h_j} \left[\sum_{h_k \ge h_j} q_{k\to[j]}^{h_k} \right] \times \left\{ \prod_{k'\in c(j)\backslash k} \left[\sum_{h_{k'}} \mathrm{e}^{-\beta E_{jk'}} q_{k'\to[j]}^{h_{k'}} \right] - (1-\delta_0^{h_j}) \times \prod_{k'\in c(j)\backslash k} \left[\sum_{h_{k'}} \mathrm{e}^{-\beta E_{jk'}} q_{k'\to[j]}^{h_{k'}} - q_{k'\to[j]}^{h_j-1} \right] \right\}, \tag{B5}$$

where z_j is expressed as

$$z_j = \sum_{h_j\ge 0} q_{j\to[j]}^{h_j} \left\{ \prod_{k'\in c(j)} \left[\sum_{h_{k'}} \mathrm{e}^{-\beta E_{jk'}} q_{k'\to[j]}^{h_{k'}} \right] - (1-\delta_0^{h_j}) \times \prod_{k'\in c(j)} \left[\sum_{h_{k'}} \mathrm{e}^{-\beta E_{jk'}} q_{k'\to[j]}^{h_{k'}} - q_{k'\to[j]}^{h_j-1} \right] \right\}. \tag{B6}$$

The mean fraction ρ_{modelR} of feedback arcs is then obtained through

$$\rho_{\mathrm{modelR}} = \frac{1}{M} \sum_{(j,k)\in G} \rho_{jk}. \tag{B7}$$

The free energy $F_{\mathrm{modelR}}(\beta) \equiv -\frac{1}{\beta}\ln Z_{\mathrm{modelR}}(\beta)$ of the whole system is computed through[23–28]

$$F_{\mathrm{modelR}}(\beta) = \sum_{j=1}^{N} \left[f_{[j]} - d_j^{\mathrm{in}} f_j \right], \tag{B8}$$

where $f_{[j]}$ is the free energy contribution of constraint $[j]$, and f_j is the free energy contribution of node j. The expressions for these two free energy contributions are, respectively,

$$f_j = -\frac{1}{\beta} \ln \left\{ \sum_{h_j} p_{[j]\to j}^{h_j} q_{j\to[j]}^{h_j} \right\}, \tag{B9a}$$

$$\begin{aligned} f_{[j]} &= -\frac{1}{\beta} \ln \left\{ \sum_{h_j} q_{j\to[j]}^{h_j} \times \sum_{\{h_k : k\in c(j)\}} \left[1 - (1-\delta_0^{h_j}) \prod_{k\in c(j)} (1-\delta_{h_k+1}^{h_j}) \right] \times \prod_{k\in c(j)} \left[\mathrm{e}^{-\beta E_{jk}} q_{k\to[j]}^{h_k} \right] \right\} \\ &= -\frac{1}{\beta} \ln \left\{ \sum_{h_j} q_{j\to[j]}^{h_j} \left[\prod_{k\in c(j)} \left[\sum_{h_k} \mathrm{e}^{-\beta E_{jk}} q_{k\to[j]}^{h_k} \right] - (1-\delta_0^{h_j}) \times \prod_{k\in c(j)} \left[\sum_{h_k} \mathrm{e}^{-\beta E_{jk}} q_{k\to[j]}^{h_k} - q_{k\to[j]}^{h_j-1} \right] \right] \right\}. \end{aligned} \tag{B9b}$$

The free energy density is then $f_{\mathrm{modelR}} \equiv \frac{1}{N} F_{\mathrm{modelR}}(\beta)$. Notice that the free energy contribution $f_{[j]}$ of a constraint $[j]$ also contains the contributions of node j and all its downstream neighbors (see Fig. B1), therefore the free energy contribution f_j of a node j has been considered $(d_j^{\mathrm{in}}+1)$ times in the first summation of Eq. (B8). The second summation of Eq. (B8) corrects this over-counting.

The entropy density s_{modelR} of the system is then

$$s_{\mathrm{modelR}} = \beta\left(\alpha\rho_{\mathrm{modelR}} - f_{\mathrm{modelR}}\right). \tag{B10}$$

This expression can be understood from the relationship that $Z_{\mathrm{modelR}} \approx \exp\left(-\beta M \rho_{\mathrm{modelR}} + N s_{\mathrm{modelR}}\right)$.

B3. Theoretical and algorithmic results on a special kind of regular random directed networks

Due to the many-body nature of the node level constraints, the above-mentioned RS mean field theory is computationally quite inefficient. As a first test of this theory, we apply it to a special kind of regular random directed networks, namely, random directed networks in which each node has the same number K of incoming arcs and the same number K of outgoing arcs. The connectivity pattern of such a random network is otherwise completely random. Such a network may

be referred to as a balanced random regular (BRR) network. For this BRR network ensemble, we can assume that the cavity probability distributions $p^h_{[j]\to j}$ are independent of the node j and are all equal to the same distribution $p^h_{\text{con-to-self}}$, and similarly all the cavity probability distributions $p^h_{[j]\to k}$ are independent of the node j and the downstream neighbor k but are equal to the same distribution $p^h_{\text{con-to-down}}$. Under these two additional assumptions the BP equation (B2) can be simplified and a fixed-point solution can be obtained by numerical iterations.

Figure B2 shows the theoretically predicted relationship between entropy density and feedback arc fraction at $K = 5$. As long as the level upper-bound $D \geq 50$, the RS theoretical results are almost independent of D. At feedback arc fraction $\rho = 0.2620$ the entropy density changes from being positive to being negative. Therefore this mean field theory predicts the minimum fraction of feedback arcs to be $\rho = 0.2620$.

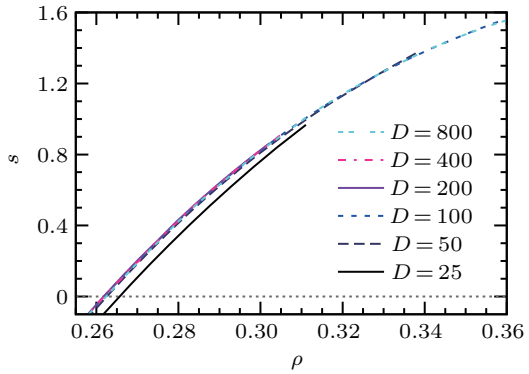


Fig. B2. (color online) Replica-symmetric mean field results (based on model-R) for a special type of regular random directed networks. Each node in the random network has $K = 5$ incoming arcs and $K = 5$ outgoing arcs. The level upper-bound D ranges from $D = 25$ to $D = 800$ in the mean-field computations. s, entropy density; ρ, fraction of feedback arcs.

We can implement a BPD message-passing algorithm based on the BP equation (B2) to construct feedback arc sets for single BRR network instances (see Appendix D). We have applied this BPD algorithm to two single network instances, one with $N = 10^4$ nodes and the other with $N = 10^5$ nodes ($K = 5$ in both cases). Figure B3 shows the averaged results over 48 independent runs of the BPD algorithm at various different values of the inverse temperature β. We notice that the BPD results obtained at $D = 50$ are slightly better than those obtained at $D = 100$; we also notice that the BPD algorithm is not sensitive to the value of the inverse temperature β as long as $\beta \geq 2$. Most strikingly, we find that the mean fraction $\rho \approx 0.243$ of feedback arcs in solutions obtained by the BPD algorithm is noticeably less than the predicted minimum fraction 0.262 by the RS mean field theory of model-R. Indeed for the studied network instance of size $N = 10^5$, the feedback arc fraction ρ in the best solution obtained by BPD is $\rho = 0.2417$. The associated BPD algorithm of the relaxed model (model-E, see Appendix C) gives very close results ($\rho \approx 0.2410$) for this network instance, while the SA algorithm (see Appendix E) achieves even better solutions with $\rho \approx 0.2212$.

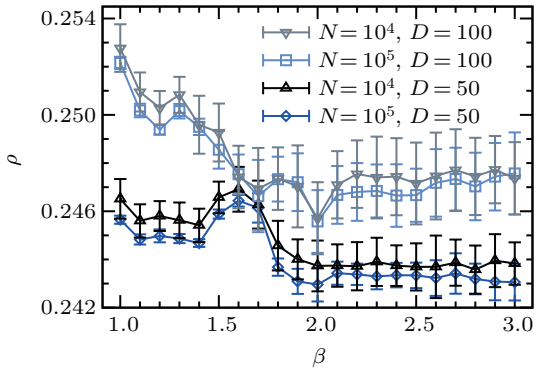


Fig. B3. (color online) Results obtained by the BPD algorithm (based on model-R) for two balanced regular random directed network instances (one of size $N = 10^4$ and the other of size $N = 10^5$), in which each node has $K = 5$ incoming arcs and $K = 5$ outgoing arcs. Each data point is averaged over 48 independent runs of the BPD algorithm (the level upper-bound is fixed to $D = 50$ or $D = 100$). ρ, fraction of feedback arcs in the constructed solutions; β, the inverse temperature of the BPD algorithm.

The fact that the theoretically predicted minimum feedback arc fraction being higher than the results obtained by the BPD and SA algorithms indicates that the level constraints of model-R are too strong for the feedback arc set problem.

Appendix C: The replica-symmetric mean-field theory for the relaxed model

Let us refer to the relaxed model as model-E (the "easy" model). There is essentially no constraint in this model except that the hierarchical level h_i of each node i should be an integer in the range $[0, D-1]$. Each directed arc (j,k) causes a two-body interaction between node j and node k, and the partition function of model-E is

$$Z_{\text{modelE}}(\beta) = \sum_{h_1=0}^{D-1} \cdots \sum_{h_N=0}^{D-1} \prod_{(j,k)\in G} \psi_{jk}(h_j, h_k), \tag{C1}$$

where $\psi_{jk}(h_j, h_k) = 1$ for $h_j > h_k$ and $\psi_{jk}(h_j, h_k) = \mathrm{e}^{-\beta}$ for $h_j \leq h_k$. The RS mean-field theory for this partition function has already been briefly described in the main text, here we add some more technical explanations.

C1. The belief-propagation equation

For an arc (i, j) from node i to node j, its probability of being a feedback arc is

$$\rho_{ij} = \frac{\mathrm{e}^{-\beta} \sum_{h_i=0}^{D-1} q^{h_i}_{i\to j} \sum_{h_j=h_i}^{D-1} q^{h_j}_{j\to i}}{1 - (1-\mathrm{e}^{-\beta}) \sum_{h_i=0}^{D-1} q^{h_i}_{i\to j} \sum_{h_j=h_i}^{D-1} q^{h_j}_{j\to i}}, \tag{C2}$$

where $q^{h_i}_{i\to j}$ denotes the cavity probability that node i will be at level h_i if node j is absent; similarly $q^{h_j}_{j\to i}$ is the cavity probability that node j will be at level h_j if node i is absent. Notice that in Eq. (C2) the product $q^{h_i}_{i\to j}q^{h_j}_{j\to i}$ is the joint probability of h_i and h_j when the arc (i,j) is absent (assuming that node i and node j are then independent); and the term $\sum_{h_i\geq 0}\sum_{h_j\geq h_i} q^{h_i}_{i\to j}q^{h_j}_{j\to i}$ is then the total probability that $h_i \leq h_j$ in the absence of the arc (i,j). The belief-propagation equation for the cavity probabilities are expressed as[23–28]

$$q^{h_j}_{j\to k} = \frac{1}{z_{j\to k}} \prod_{i\in p(j)} \left[\mathrm{e}^{-\beta} + (1-\mathrm{e}^{-\beta}) \sum_{h_i=h_j+1}^{D-1} q^{h_i}_{i\to j} \right] \times \prod_{k'\in c(j)\backslash k} \left[\mathrm{e}^{-\beta} + (1-\mathrm{e}^{-\beta}) \sum_{h_{k'}=0}^{h_j} q^{h_{k'}}_{k'\to j} \right], \tag{C3a}$$

$$q^{h_j}_{j\to i} = \frac{1}{z_{j\to i}} \prod_{i'\in p(j)\backslash i} \left[\mathrm{e}^{-\beta} + (1-\mathrm{e}^{-\beta}) \sum_{h_{i'}=h_j+1}^{D-1} q^{h_{i'}}_{i'\to j} \right] \times \prod_{k\in c(j)} \left[\mathrm{e}^{-\beta} + (1-\mathrm{e}^{-\beta}) \sum_{h_k=0}^{h_j} q^{h_k}_{k\to j} \right], \tag{C3b}$$

where node k in Eq. (C3a) belongs to set $c(j)$ and node i in Eq. (C3b) belongs to set $p(j)$; $z_{j\to k}$ and $z_{j\to i}$ are two probability normalization constants. Notice that Eq. (C3) is equivalent to Eq. (4) of the main text.

C2. Thermodynamic quantities

At a given value of the inverse temperature β, we can compute the mean fraction of feedback arcs as

$$\rho_{\mathrm{modelE}} = \frac{1}{M} \sum_{(i,j)\in G} \rho_{ij}. \tag{C4}$$

The total free energy of model-E, defined by $F_{\mathrm{modelE}} = -\frac{1}{\beta}\ln Z_{\mathrm{modelE}}(\beta)$, can be expressed as[23–28]

$$F_{\mathrm{modelE}} = \sum_{j=1}^{N} f_j - \sum_{(k,l)\in G} f_{kl}, \tag{C5}$$

where f_j and f_{kl} are, respectively, the node and arc contribution to the free energy:

$$f_j = -\frac{1}{\beta}\ln\left\{ \sum_{h_j=0}^{D-1} \prod_{i\in p(j)} \left[1-(1-\mathrm{e}^{-\beta}) \sum_{h_i=0}^{h_j} q^{h_i}_{i\to j} \right] \times \prod_{k\in c(j)} \left[1-(1-\mathrm{e}^{-\beta}) \sum_{h_k=h_j}^{D-1} q^{h_k}_{k\to j} \right] \right\}, \tag{C6a}$$

$$f_{kl} = -\frac{1}{\beta}\ln\left\{ 1-(1-\mathrm{e}^{-\beta}) \sum_{h_k=0}^{D-1}\sum_{h_l=h_k}^{D-1} q^{h_k}_{k\to l} q^{h_l}_{l\to k} \right\}. \tag{C6b}$$

To understand Eq. (C5) in an intuitive way, we notice that the free energy contribution f_j of each node j includes the contributions from all the attached outgoing and incoming arcs, therefore the contribution of an arc (j,k) is considered twice (in f_j and f_k); such an over-counting is corrected by the second summation of Eq. (C6).

The free energy density is simply $f_{\mathrm{modelE}} \equiv \frac{1}{N}F_{\mathrm{modelE}}$. And the entropy density s_{modelE} at a given value of β is then evaluated as

$$s_{\mathrm{modelE}} = \beta\left[\alpha\rho_{\mathrm{modelE}} - f_{\mathrm{modelE}}\right], \tag{C7}$$

where α is the arc density. Equation (C7) is justified by the fact that $Z_{\mathrm{modelB}} \approx \exp\left(-M\beta\rho_{\mathrm{modelE}} + Ns_{\mathrm{modelE}}\right)$.

C3. Computation for single network instances and for an ensemble of networks

The RS mean-field theory can be applied on single instances of directed networks. Given a directed network G, we first iterate the BP equation (C3) on all the directed arcs a number t_0 (e.g., $t_0 = 200$) of times to reach a fixed point or to bring the set of all the cavity probability distributions $\{q^{h_j}_{j\to i}, q^{h_j}_{j\to k}\}$ close to a steady state. Then we repeat the BP iteration an additional number t_1 (e.g., $t_1 = 1000$) of times, at each time step we compute all the node free energy contributions f_j, all the arc free energy contributions f_{kl} and probabilities ρ_{kl} to evaluate the values of f_{modelE}, ρ_{modelE}, and s_{modelE}. The averaged results of f_{modelE}, ρ_{modelE}, and s_{modelE} over these t_1 iterations are then reported as the free energy density, the mean fraction of feedback arcs, and the entropy density, respectively.

Some BP simulation results obtained on Erdös–Rényi (ER) random directed networks of arc density $\alpha = 5.0$ are shown Fig. C1.

We can also obtain ensemble-averaged results for random directed networks which have no any correlation in the connectivity pattern. For such networks, let us denote by $P(d^{\mathrm{in}}, d^{\mathrm{out}})$ the joint probability that a randomly chosen node has d^{in} incoming arcs and d^{out} outgoing arcs. We construct two large sets $\mathcal{M}_{\mathrm{p-to-c}}$ and $\mathcal{M}_{\mathrm{c-to-p}}$, with $\mathcal{M}_{\mathrm{p-to-c}}$ storing many (e.g., $L = 10^4$) cavity probability distributions $q^{h_i}_{i\to j}$ and $\mathcal{M}_{\mathrm{c-to-p}}$ storing many cavity probability distributions $q^{h_j}_{j\to i}$ for different arcs (i,j). We then update these two sets a large number of times (e.g., $10^4 \times L$) to drive these two sets to a steady state and to evaluate the ensemble-averaged values of the thermodynamic quantities. At each updating process, two non-negative integers d^{in} and d^{out} are drawn from the joint distribution $P(d^{\mathrm{in}}, d^{\mathrm{out}})$ and assigned to a node (say j) as its in-degree and out-degree, respectively. The d^{in} cavity probability distributions $q^{h_i}_{i\to j}$ from the incoming arcs (i,j) are then drawn from the set $\mathcal{M}_{\mathrm{p-to-c}}$ uniformly at random and with replacement; similarly the d^{out} cavity probability distributions $q^{h_k}_{k\to j}$ from the outgoing arcs (j,k) are drawn from the set $\mathcal{M}_{\mathrm{c-to-p}}$ uniformly at random and with replacement. Then d^{in} new cavity probability distributions $q^{h_j}_{j\to i}$ and d^{out} new cavity probability distributions $q^{h_j}_{j\to k}$ are computed according to Eq. (C3), and they replace d^{in} randomly chosen elements of set $\mathcal{M}_{c-to-p}$ and d^{out} randomly chosen elements of set $\mathcal{M}_{p-to-c}$, respectively. The values of f_j (for node j) and f_{jk} and f_{ij} and ρ_{jk} and ρ_{ij} for all the attached arcs of node j are computed during this process.

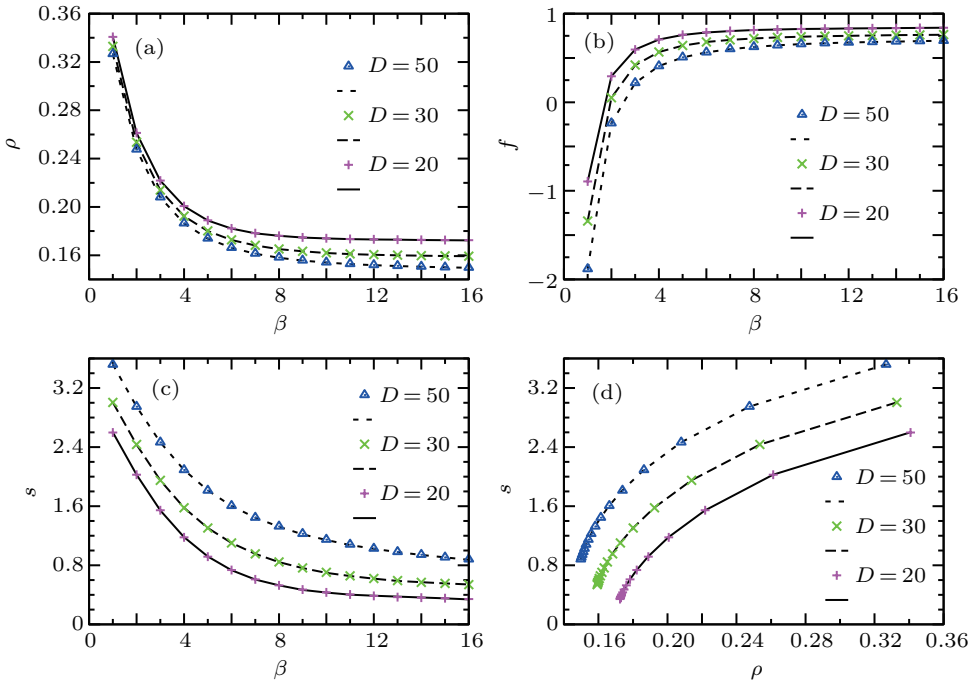


Fig. C1. (color online) Some RS mean-field theoretical results obtained on model-E (C1) for ER random directed networks of arc density $\alpha = 5.0$. The dots are the averaged simulation results obtained by BP iterations on 40 independent ER network instances of $N = 10^5$, with the level upper-bound being $D = 20$ (pluses), $D = 30$ (crosses), and $D = 50$ (triangles). The corresponding lines are the results obtained by RS population dynamics (which corresponds to network size $N = \infty$). (a) the fraction ρ of feedback arcs as a function of inverse temperature β; (b) the free energy density f as a function of β; (c) the entropy density s as a function of β; (d) the entropy density s as a function of ρ, obtained by eleminating β from data in panels (a) and (c).

As demonstrated in Fig. C1, the results obtained by this RS population dynamics simulation are in complete agreement with the BP results obtained on single network instances.

Appendix D: Minimizing the number of feedback arcs by belief-propagation–guided decimation

We have implemented two versions of the BPD algorithm based on the strongly constrained model-R (Appendix B) and on the relaxed model-E (Appendix C), respectively. These two versions of the BPD algorithm have the same algorithmic design and the same overall structure except for the differences in the adopted BP equations. Here we describe some technical details of the algorithm.

Given an input directed network G, we first simplify it by recursively removing all the nodes which have no outgoing or incoming arcs. The arcs attached to these removed nodes are also deleted from the network. Then every node in the remaining subnetwork must have both incoming and outgoing arcs. The BPD search process is then applied on this subnetwork.

Starting from an initially empty feedback arc set Λ, in each decimation step of the BPD algorithm: 1) the BP equation is iterated on all the remaining arcs of the network G for a number r of times (e.g., $r = 10$); 2) then the probability ρ_{ij} for each remaining arc (i, j) to be a feedback arc is estimated based on the RS mean-field formula; 3) then a tiny fraction ε of the remaining arcs (e.g., $\varepsilon = 0.01$ or $\varepsilon = 0.005$) which have the largest estimated feedback probabilities are deleted from the network and are added to the set Λ; 4) then the network is further simplified by recursively deleting all the nodes which have no outgoing arc or have no incoming arc.

When there is no directed cycle in the remaining network, the BPD process will terminate. Then we check every arc (i, j) in the set Λ in a random order and delete it from Λ if and only if the reduced set Λ is still a feedback arc set. The final set Λ is then reported.

For random directed networks, our empirical results suggest that the BPD algorithm based on the relaxed model-E is much faster and also achieves slightly better solutions when compared with the BPD algorithm based on the strongly restricted model-R.

Appendix E: Minimizing the number of feedback arcs by simulated annealing

To perform simulated annealing on an input directed network G, we need to initialize the node permutation $\mathcal{P} = (v_1, v_2, \ldots, v_N)^{\mathrm{T}}$ in a proper way. For this purpose, we first find all the strongly connected components (SCCs) of this network; and then we construct an initial node permutation $\mathcal{P}$ in the following iterative way: 1) randomly choose a SCC which receives no incoming arcs from other SCCs (there must be at least one such SCC); 2) put the nodes of this SCC to the not-yet occupied top positions of $\mathcal{P}$ in a random order; 3) then delete this SCC and all its outgoing arcs to other SCCs; 4) re-

peat the preceding three steps on another remaining SCC as long as the directed graph is not yet empty.

We illustrate in Fig. E1 the two basic updating rules of the SA algorithm, which were inspired by the earlier work of Ref. [30]. This figure complements the descriptions in the main text. In our actual implementation of the SA algorithm, to select a feedback arc most efficiently for the action shown in Fig. E1(b), we store all the feeback arcs in different lists: the feedback arcs (i,j) whose associated energy changes $s_{i,j}^{i\uparrow} \leq 0$ are all stored in the list U_0; the feedback arcs (i,j) whose associated energy changes $s_{i,j}^{i\uparrow} = n\ (\geq 1)$ are all stored in the list U_n. To perform the updating shown in Fig. E1(b), we first choose an integer value $n \geq 0$ according to the probability

$$P_{up}(n) = \frac{|U_n|\,\mathrm{e}^{-\beta n}}{\sum\limits_{n'\geq 0} |U_{n'}|\,\mathrm{e}^{-\beta n'}}, \tag{E1}$$

where $|U_n|$ denotes the length of the list U_n; then we choose an arc (i,j) in the list U_n uniformly at random and move node i to be immediately above node j in the permutation $\mathcal{P}$.

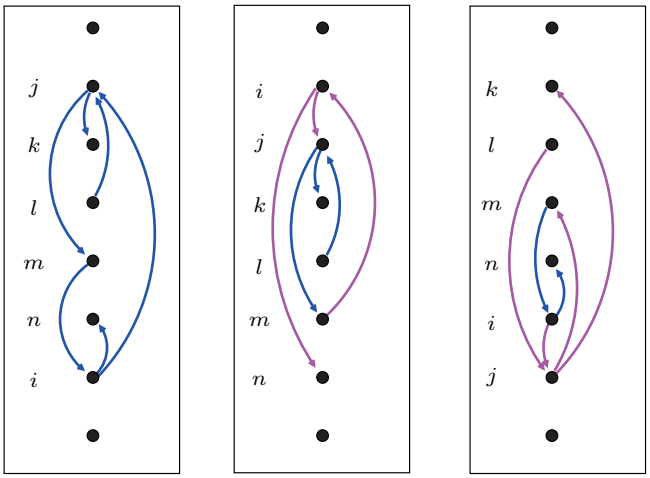


Fig. E1. (color online) Changing the vertical positions of the nodes by simulated annealing to reduce the number of feedback (upward) arcs. In this example, node i has two outgoing arcs (i,j) and (i,n) and one incoming arc (m,i), while node j has two outgoing arcs (j,k) and (j,m) and two incoming arcs (i,j) and (l,j). (a) The node order before updating (only the nodes i, j and their incoming and outgoing neighbors are shown here for clarity). (b) Node i is moved to the position immediately above node j; the change in the number of feedback arcs is $s_{i,j}^{i\uparrow} = -1$, since now (i,j) and (i,n) both become feedforward (downward) arcs and (m,i) changes to be a feedback arc. (c) Node j is moved to the position immediately below node i; the change in the number of feedback arcs is $s_{i,j}^{j\downarrow} = 0$, since now (i,j) and (l,j) change to be feedforward arcs and (j,k) and (j,m) change to be feedback arcs.

Similarly, to speed up the downward updating shown in Fig. E1(c), we also store all the feedback arcs in another set of lists: the feedback arcs (i,j) whose associated energy changes $s_{i,j}^{j\downarrow} \leq 0$ are all stored in the list D_0; the feedback arcs (i,j) whose associated energy changes $s_{i,j}^{j\downarrow} = n\ (\geq 1)$ are all stored in the list D_n. A downward updating is achieved in two steps: first, an integer value $n \geq 0$ is chosen according to the probability

$$P_{\mathrm{down}}(n) = \frac{|D_n|\,\mathrm{e}^{-\beta n}}{\sum\limits_{n'\geq 0} |D_{n'}|\,\mathrm{e}^{-\beta n'}}, \tag{E2}$$

where $|D_n|$ denotes the length of the list D_n; second, an arc (i,j) in the list D_n is randomly chosen and node j is moved to be immediately below node i in the permutation $\mathcal{P}$.

Notice that, after each elementary updating of the node order, the lists U_0, U_1, ..., D_0, D_1, ... should be updated if necessary. These can be done very efficiently, since only the attached incoming and outgoing arcs of the nodes i and j of the chosen arc (i,j) need to be considered.

At each inverse temperature β, a total number of $2c_0N$ elementary updates are carried out, with one half of them being the upward movements shown in Fig. E1(b) and the other half being the downward movements shown in Fig. E1(c). The parameter c_0 is chosen in the range $c_0 \in [5, 10000]$ (the default value is $c_0 = 5$). Larger values of c_0 usually lead to slightly improved solutions, but the simulation times are longer.

The inverse temperature β then increases by a factor of $1/c_1$ (c_1 is set to be 0.99 in the present work) after $2c_0N$ elementary updates have been carried out. If at c_2 consecutive values of β (with $c_2 = 50$) the SA search process fails to find a node permutation whose corresponding number of feedback (upward) arcs is smaller than that of any previously visited node permutation, the search is then terminated and the best solution reached during the whole SA process is reported.

Appendix F: Random network instances

We generate directed random networks following conventional methods in the literature.

To generate a directed random Erdös–Rényi (ER) network G, we start from an empty network containing N nodes and then add arcs to the network one by one until the total number of added arcs reaches the specified value M. The two end nodes i and j of each candidate arc (i,j) are chosen uniformly at random from the N nodes, and this candidate arc is accepted if it has yet been added to the network and $i \neq j$.

On the other hand, to generate a directed regular random (RR) network of arc density α, we first assign to each node $d = 2\alpha$ half-edges (d must be an integer), and then repeat the process of glueing two randomly chosen half-edges into a complete edge between two nodes and then assigning a random direction to this edge (self-connections and multiple arcs between the same pair of nodes are not allowed).

To generate a directed random scale-free network according to the configurational model (i.e., a SFC network), we first assign to each node i d_i^{in} incoming half-arcs and d_i^{out} outgoing half-arcs, with d_i^{in} and d_i^{out} being independent random integers generated according to the in-degree power-law distribution

$P_{\rm in}(d)$ and out-degree power-law distribution $P_{\rm out}(d)$, respectively. The expressions for these two degree distributions are

$$P_{\rm in}(d) = \frac{d^{-\gamma_{\rm in}}}{\sum_{d'=d_{\rm min}}^{d_{\rm max}} (d')^{-\gamma_{\rm in}}}, \quad (d_{\rm min} \le d \le d_{\rm max}), \tag{F1a}$$

$$P_{\rm out}(d) = \frac{d^{-\gamma_{\rm out}}}{\sum_{d'=d_{\rm min}}^{d_{\rm max}} (d')^{-\gamma_{\rm out}}}, \quad (d_{\rm min} \le d \le d_{\rm max}), \tag{F1b}$$

where $\gamma_{\rm in}$ is the in-degree decay exponent and $\gamma_{\rm out}$ is the out-degree decay exponent; the parameter $d_{\rm min}$ is the minimum value of degree, and $d_{\rm max}$ is the maximum value of degree. We set $d_{\rm min} = 2$ and $d_{\rm max} = \sqrt{N}$ in this study. After the in- and out-degree for each node are assigned, if the sum $M_{\rm in} \equiv \sum_{i=1}^{N} d_i^{\rm in}$ is larger (respectively, smaller) than the sum $M_{\rm out} \equiv \sum_{i=1}^{N} d_i^{\rm out}$, we then change $|(M_{\rm in} - M_{\rm out})|/2$ randomly chosen incoming (respectively, outgoing) half-arcs into outgoing (respectively, incoming) half-arcs to make $M_{\rm in} = M_{\rm out}$. Finally we repeat the process of glueing one randomly chosen outgoing half-arc with one randomly chosen incoming half-arc to form a complete arc (self-connections and multiple arcs between the same pair of nodes are not allowed).

To generate a directed random scale-free network according to the static model (i.e., a SFS network[33]), we proceed as follows: 1) randomly permute the N nodes and then assign the r-th node (say node i) in this permutation the weight $w_i^{\rm in} = r^{-1/(\gamma_{\rm in}-1)}$; 2) repeatedly add an incoming half-arc to the network and attach it to a node i with probability proportional to its assigned weight $w_i^{\rm in}$; 3) randomly permute the N nodes again and then assign the r-th node (say node i') in this new permutation the weight $w_{i'}^{\rm out} = r^{-1/(\gamma_{\rm out}-1)}$; 4) repeatedly add an outgoing half-arc to the network and attach it to a node j with probability proportional to its assigned weight $w_j^{\rm out}$; 5) repeatedly glueing one randomly chosen outgoing half-arc with a randomly chosen incoming half-arc to form a full arc (self-connections and multiple arcs between the same pair of nodes are not allowed). After this directed network is generated, the probability that a randomly chosen node to have d incoming arcs is $P_{\rm in}(d) \propto d^{-\gamma_{\rm in}}$ for large values of d, and the probability that a randomly chosen node to have d outgoing arcs is $P_{\rm out}(d) \propto d^{-\gamma_{\rm out}}$ for large values of d.[33]

Appendix G: Visualizing a strongly connected network component

Let us denote by $G_{\rm SCC}$ a strongly connected component of a network G and assume that this component has n nodes. To visualize this strongly connected component, we need to specify for each node a two-dimensional coordinate (x, y). Our recipe for this task proceeds as follows:

1. Run the SA algorithm or the BPD algorithm on $G_{\rm SCC}$ to obtain a near-minimum feedback arc set Λ.

2. Then construct a node hierarchy based on set Λ according to the simple procedure of Appendix A; set the y-coordinate y_i of each node i to be identical to its hierarchical level h_i.

3. Then randomly permute the n nodes and assign the r-th node (say i) in this permutation the x-coordinate $x_i = r$.

4. Then try to exchange the x-coordinates of the n nodes to minimize the total "connection cost" $C(x_1, x_2, \ldots, x_n)$ defined by

$$C(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in G_{\rm SCC}} |x_i - x_j|. \tag{G1}$$

We can simply use simulated annealing to minimize this connection cost: In each elementary updating step, a proposal is made to exchange the x-coordinates of two nodes; if the change in the connection cost ΔC is zero or negative, this proposal is accepted, otherwise it is accepted with probability $\mathrm{e}^{-\Delta C/T}$, with the parameter T (the "temperature") being a slowly decreasing quantity.

5. After a near-minimum x-coordinate solution has been obtained for the connection cost $C(x_1, \ldots, x_n)$, then draw the nodes and arcs of the strongly connected component $G_{\rm SCC}$ according to the determined (x, y) coordinates of all the n nodes.

The motivation for us to minimize the connection cost (G1) is simple: we want to make the densely connected nodes to form clusters along the horizontal (x) direction, so that possible community structures with the strongly connected network component can be clearly visualized, besides the hierarchical structure along the vertical (y) direction.

Figure 1 of the main text was drawn by the above-mentioned method. Notice that our method can also be applied on the whole directed network G: we can first work on the individual strongly connected components and then combine all these components to form a two-dimensional plot for the whole directed network.

References

[1] White J G, Southgate E, Thomson J N and Brenner S 1986 *Phil. Trans. R. Soc. Lond. B* **314** 1
[2] Li F, Long T, Lu Y, Ouyang Q and Tang C 2004 *Proc. Natl. Acad. Sci. USA* **101** 4781
[3] Oda K, Matsuoka Y, Funahashi A and Kitano H 2005 *Mol. Syst. Biol.* **1** 2005.0010
[4] Alon U 2007 *Nature Rev. Genetics* **8** 450
[5] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D and Alon U 2002 *Science* **298** 824
[6] Leicht E A and Newman M E J 2008 *Phys. Rev. Lett.* **100** 118703
[7] Fortunato S 2010 *Phys. Rep.* **486** 75
[8] Tarjan R E 1972 *SIAM J. Comput.* **1** 146
[9] Dorogovtsev S N, Mendes J F F and Samukhin A N 2001 *Phys. Rev. E* **64** 025101(R)
[10] Jeong H, Tombor B, Albert R, Oltvai Z N and Barabási A L 2000 *Nature* **407** 651
[11] Ravasz E, Somera A L, Mongru D A, Oltvai Z N and Barabási A L 2002 *Science* **297** 1551
[12] Lan Y and Mezić I 2011 *BMC Syst. Biol.* **5** 37

[13] Corominas-Murtra B, Goñi J, Solé R V and Rodríguez-Caso C 2013 *Proc. Natl. Acad. Sci. USA* **110** 13316

[14] Domínguez-García V, Pigolotti S and Muñoz M A 2014 *Sci. Rep.* **4** 7497

[15] Fiedler B, Mochizuki A, Kurosawa G and Saito D 2013 *J. Dynam. Differ. Equat.* **25** 563

[16] Xu J and Lan Y 2015 *PLoS ONE* **10** e0125886

[17] Ispolatov I and Maslov S 2008 *BMC Bioinformatics* **9** 424

[18] Gupte M, Shankar P, Li J, Muthukrishnan S and Iftode L 2011 *Proceedings of the twentieth International World Wide Web Conference*, March 28–April 01, 2011, Hyderabad, India, p. 557

[19] Ulanowicz R E, Bondavalli C and Egnotovich M S 1998 *Network analysis of trophic dynamics in south Florida ecosystem, fy* 97*: The Florida Bay ecosystem. Technical report, Chesapeake Biological Laboratory, Solomons*, 1998

[20] Liu Y Y Barabśi A L 2016 *Rev. Mod. Phys.* **88** 035006

[21] Eades P and Sugiyama K 1990 *J. Information Processing* **13** 424

[22] Garey M and Johnson D S 1979 *Computers and Intractability: A Guide to the Theory of NP-Completeness* (San Francisco: Freeman)

[23] Mézard M and Montanari A 2009 *Information, Physics, and Computation* (New York: Oxford Univ. Press)

[24] Mézard M and Parisi G 2001 *Eur. Phys. J. B* **20** 217

[25] Bayati M, Borgs C, Braunstein A, Chayes J, Ramezanpour R and Zecchina R 2008 *Phys. Rev. Lett.* **101** 037208

[26] Altarelli F, Braunstein A, Dall'Asta L and Zecchina R 2013 *J. Stat. Mech.: Theor. Exp.* **2013** P09011

[27] Guggiola A and Semerjian G 2015 *J. Stat. Phys.* **158** 300

[28] Zhou H J 2016 *J. Stat. Mech.: Theor. Exp.* **2016** 073303

[29] Kirkpatrick S, Gelatt Jr. C D and Vecchi M P 1983 *Science* **220** 671

[30] Galinier P, Lemamou E and Bouzidi M W 2013 *J. Heuristics* **19** 797

[31] Qin S M and Zhou H J 2014 *Eur. Phys. J. B* **87** 273

[32] Dorogovtsev S N and Mendes J F F 2002 *Adv. Phys.* **51** 1079

[33] Goh K I, Kahng B and Kim D 2001 *Phys. Rev. Lett.* **87** 278701

[34] Pardalos P M, Qian T B and Resende M G C 1998 *J. Combin. Optim.* **2** 399

[35] Leskovec J, Huttenlocher D and Kleinberg J 2010 *Proceedings of the 19th International Conference on World Wide Web*, April 26-30, 2010, Raleigh, NC, USA, p. 641

[36] Ripeanu M, Foster I and Iamnitchi A 2002 *IEEE Internet Comput.* **6** 50

[37] Mugisha S and Zhou H J 2016 *Phys. Rev. E* **94** 012305

[38] Braunstein A, Dall'Asta L, Semerjian G and Zdeborová L 2016 *Proc. Natl. Acad. Sci. USA* **113** 12368