Spectral reconstruction of networks using combinatorial optimization algorithms *

Francesc Comellas and Jordi Diaz-Lopez

Departament de Matemàtica Aplicada IV, Universitat Politècnica de Catalunya
Avda. Canal Olímpic s/n, 08860 Castelldefels, Catalonia, Spain
comellas@ma4.upc.edu
http://www-ma4.upc.edu/~comellas/

Abstract. In this paper we study the reconstruction of a network topology from the eigenvalues of its Laplacian matrix. We introduce a new simple cost function and consider three combinatorial optimization methods - simulated annealing, tabu search, and multiagent optimization (ants)- while comparing their performance when reconstructing different categories of networks –random, regular, small-world, scale-free and clustered– from their eigenvalues. We show that tabu search provides more accurate reconstructions than the other methods, while all the algorithms considered allow an exact reconstruction of small networks and lead to good approximations in the case of networks with larger orders.

1 Introduction

In recent years there has been a growing interest in the study of complex networks, related to transportation and communication systems (WWW, Internet, power grid, etc.) Many of these networks are very large with a number of nodes very often in the thousands. To store the topological details of the network requires knowing the list of adjacencies and, although usually the networks are sparse, this means the use of a large amount of memory. In contrast, the spectrum of the network (eigenvalues of the associated adjacency or Laplacian matrix) contains important information with significantly less memory use. Therefore it is of interest to reconstruct, even partially, a network from its spectrum. In [14], Ipsen and Mikhailov use simulated annealing with an elaborate cost function based on the spectral density to perform such a reconstruction. Here we propose a simple cost function and compare its performance using simulated annealing and two other combinatorial optimization algorithms. The methods are probabilistic, i.e. they contain a random component, and as a consequence we can not guarantee that the algorithm will find an optimal reconstruction, but we show that the final networks match the originals in their main topological properties.

In the next section, we introduce the mathematical notation and concepts necessary for our study. Section 3 contains a description of the combinatorial

* Research supported by the Ministerio de Educación y Ciencia, Spain, and the European Regional Development Fund under project TEC2005-03575.

optimization algorithms considered and our main results are presented in Section 4.

2 The Laplacian spectrum of a graph and its reconstruction

Let us consider the Laplacian of the graph $G = G(V, E)$, with vertex set V (order $n = |V|$) and edge set E, associated with a network. The Laplacian is a symmetric matrix with zero row-sums that accounts for the topology of the network, defined to be $L_{ij} = -1$ if nodes i and j are connected, $L_{ii} = \delta_i$ if node i has degree δ_i (i.e. is connected to δ_i other nodes), and $L_{ij} = 0$ otherwise. The Laplacian matrix can be related to the adjacency matrix A of G by $L = D - A$, where D is the diagonal matrix of vertex degrees of G.

The (Laplacian) spectrum of G consists of the n eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of the Laplacian matrix and they satisfy $0 = \lambda_1 \leq \lambda_2 \leq \ldots, \leq \lambda_n$. We have also considered the reconstruction of a network from the spectra of the signless Laplacian matrix, defined as $|L| = D + A$, and the normalized Laplacian matrix, defined to be $\bar{L}_{ij} = 1/\sqrt{\delta_i \delta_j}$ if nodes i and j are connected, $\bar{L}_{ii} = 1$ if node i has degree $\delta_i > 0$, and $\bar{L}_{ij} = 0$ otherwise. For this last matrix, the eigenvalues satisfy $0 = \lambda_1 \leq \lambda_2 \leq \ldots, \leq \lambda_n \leq 2$ and $1 < \lambda_2 \leq 2$. In this paper, when we refer to the Laplacian matrix we mean the first definition unless stated otherwise. The spectrum of a graph is important as it provides bounds on its diameter, maximum and minimum degrees, and gives information about possible partitions etc. It can also be used to count the number of paths of a given length in the network, number of triangles, total number of links, etc., see [8,5]. Dynamic properties of a network, like its synchronizability, can also be determined from the eigenvalues, see [4,6]

We note that two isomorphic graphs have the same spectrum, independent of the labeling of the vertices, but there also exist non-isomorphic graphs (topologically different) with the same spectrum, known as cospectral graphs. For $n < 6$ there are no connected cospectral graphs with respect to the usual Laplacian matrix. For $n = 6$ there exist four pairs, there are 130 pairs for $n = 7$, 1767 pairs for $n = 8$, etc. The number of cospectral graphs increases rapidly with the order of the graph, but the fraction is very small (e.g. it is 0.09 for $n = 11$) and approaches zero as the order of the graph increases, see [9,13]. Hence, two graphs with the same spectrum would indeed be isomorphic with a high probability.

In this paper, we study the reconstruction of graphs from their Laplacian spectra. Note that the number of different graphs of a given order n is large even for relatively small order. For example, for $n = 40$ there are approximately 10^{186} graphs. It makes no sense to check all of these graphs to find one matching spectrum, even in an approximate way. We are in the classical situation where combinatorial optimization algorithms are useful.

The generic process is as follows: we will reconstruct a given reference graph G_0 from its spectrum $\{\lambda_i, 0 \leq i \leq n\}$. In the reconstruction process, we generate an initial random graph G_i with n vertices and we apply a process of evolution

and selection, which depends on the precise combinatorial optimization method chosen. Evolution is done by random modifications of the pattern of connections whereas the selection process is based on the spectral distance between two graphs, a concept which we discuss below.

A typical modification of a graph starts by deleting all connections of a randomly chosen vertex i. A new degree for this particular vertex is then chosen at random between 1 and $n-1$ followed by a random generation of k_i new connections.

To decide if the changes should be accepted (that is, to perform *selection*), we need a measure (cost function) of the "distance" of a given graph from the reference graph G_0. This measure is given by a spectral distance ϵ. In this paper we introduce a simple spectral distance based on the quadratic difference of the eigenvalues $\epsilon = \sum_{i=0}^{n}(\lambda_i^0 - \lambda_i^{rec})$. We have tested other spectral distances which give different weights to the eigenvalues, but they are more complex and their efficiency is similar. Also, our new distance is much more simple than the distance considered in the related study by Ipsen and Mikhailov [14]. These authors use the distance function $\epsilon = \sqrt{\int_0^\infty [\rho(\omega) - \rho_0(\omega)]^2 d\omega}$, where $\rho(\omega)$ is the spectral density defined as $\rho(\omega) = K \sum_{k=1}^{n-1} \gamma/((\omega - \omega_k)^2 + \gamma^2)$ with $\omega_k^2 = -\lambda_k$, K is a normalization constant and γ is the width of the Lorentz distribution.

The main problem with the reconstruction of a graph is to relate the generated graph to the reference graph. The graphs can be isomorphic but with permuted vertices or non-isomorphic with topological similarity that might not be manifest. As in [14], we check the similarity between two graphs in terms of the singular value decomposition [17] of their adjacency matrices. We recall that a matrix A can be decomposed into two matrices U and V and a diagonal singular value matrix Σ which satisfy $A = U\Sigma V^T$ and $\Sigma = U^T A V$. For any two graphs G_1 and G_2 with adjacency matrices A_1 and A_2, consider the function $F = F(A_1, A_2) = U_1 \Sigma_2 V_1^T = U_1 U_2^T A_2 V_2 V_1^T$ which is constructed from the singular vectors of G_1 and G_2. If the two graphs are isomorphic and their adjacency matrices only differ because of a different ordering of the vertices, it will happen that $A_1 = F(A_1, A_2)$. However, if the two graphs are not isomorphic, F will have real values not far from the values of A_1. Therefore, it is possible to define $\Delta = A_1 - F$ and use the norm $\delta = \sqrt{\sum_{i,j} \Delta_{ij}^2}/n$ to measure similarity between the graphs.

3 Combinatorial optimization algorithms

When exact methods are not possible, sometimes it is sufficient to obtain an approximate solution with a fast easily implemented method. These methods include simulated annealing, genetic algorithms, tabu search, ant colony based systems, and many other combinatorial optimization techniques.

To implement any of these optimization methods, we need a way to encode the problem which has to be solved, and a measure to quantify the "goodness" of a solution. In the case of a spectral recconstruction of a network, a possible

solution is given by the adjacency list of the graph. As described in the previous section, the solution is changed by deleting all links for a node and introducing new ones. The cost function simply measures the "distance" of the spectrum the new graph with the target spectrum.

3.1 Algorithms

We have considered three algorithms: simulated annealing, tabu search and multiagent systems (ants).

Simulated annealing. The simulated annealing (SA) technique comes from the similarity between the states of a physical system, e.g. a liquid, and the configurations of a system in a combinatorial optimization problem. If the temperature of the interacting molecules in a liquid is suddenly reduced below its freezing point, the result is a disordered glassy state with an energy higher than the true crystalline ground state. In fact the molecules are in a local energy minimum. On the other hand, if the temperature of the liquid is reduced slowly (annealing), waiting for equilibrium to be reached before a new reduction is made, the liquid freezes to the solid state through a cooling process that leads to the crystalline state, which is the global energy minimum. This scheme was first used computationally by Metropolis et al. in 1953 (then the name Metropolis algorithm was used in statistical mechanics) and in complex combinatorial optimization in 1983 [15].

In the combinatorial optimization problem, the parameters being varied are equated with atomic positions in the liquid and its energy is identified with the cost function being optimized. The temperature, T, is then defined as a control parameter related to the probability of accepting changes that worsen the state of the system, thus ensuring a more comprehensive search of the state space. Hence, in the simulated annealing, a change of state that decreases the cost ϵ is always accepted, whereas if the cost increases, the change is accepted with a certain probability $e^{-\Delta\epsilon/T}$. At a given temperature, a large number of attempts is performed to obtain a good statistical set of trials and then the temperature is decreased. This process is repeated and the system is gradually cooled until it is stopped according to some criteria such as a small number of changes accepted and/or a non-significant reduction of the cost.

`Simulated Annealing`

1. Generate an initial random graph. Fix the initial value of T and T_{min}.
2. Repeat N times.
   (a) Modify the graph and find new cost.
   (b) If better, accept it as current solution.
   (c) If worse, accept only if $e^{-\Delta\epsilon/KT} > rand()$
3. Lower T and repeat 2 until $T < T_{min}$

Tabu search. The tabu search method was described in its current form by Glover in 1986 [12]. The overall approach is to avoid cycles in the solution space by forbidding or penalizing moves that would take the current solution to others

previously visited. This ensures that new regions of the solution space will be investigated.

The tabu search starts by finding a local minimum. To avoid retracing the steps used, the method records recent moves in a list (tabu list). The tabu lists form the tabu search memory. The role of the memory can change as the algorithm proceeds. At initialization the goal is to make a coarse examination of the solution space, known as "diversification", but as candidate locations are identified the search becomes more focused to produce locally optimal solutions in a process known as "intensification".

```
Tabu search
```
1. Generate an initial random graph. Tabu list empty.
2. Repeat until stop criterion
   (a) Select vertex at random and modify an edge not in tabu list. Compute cost.
   (b) If better, accept graph and add edge to tabu list.
   (c) If worse, undo modification and add edge to tabu list
   (d) Delete old items from tabu list.

Multiagent systems (ants). The last algorithm that we have considered is *ants*, a multiagent system in which several autonomous agents work together to obtain good solutions of the problem. The *ants* algorithm is based on the idea of parallel search. Unlike other algorithms with a similar name which are generically known as ant-colony optimization, our algorithm does not use "pheromones" or local memory. Thus, it is faster and easier to implement. A version of the algorithm for the frequency assignment problem can be found in [2]. Another version for the k-partitioning of a graph has been published in [7].

The mechanism of the algorithm is as follows: We generate a random graph with n vertices. A given number of agents, which we call ants, is placed on the vertices at random. Then the ants move around the graph and change the edges according to a local optimization criterion: at a given iteration each ant moves from the current position to the adjacent vertex with the lowest local cost and performs a local optimization (deleting all edges and introducing new connections in such a way that the local cost is lowered). This local cost to be minimized is computed from the eigenvalues and eigenvectors of the adjacency matrix as follows: $C_{\text{local}} = |\lambda v_{opt} - \sum_{i \neq opt} x_i v_i|$, where v_i is the element of the eigenvector associated with the eigenvalue λ corresponding to vertex i, v_{opt} is the element associated with the current vertex being optimized, and x_i takes values 0 or 1. The values of i such that $x_i = 1$ such that the local cost is minimized will we used to reconnect the vertex considered. To perform this minimization, a simulated annealing process is used. Note that this local optimization process introduces more information (given by the eigenvectors) and complexity to the algorithm. However, we run all of the different algorithms for the same time to compare them.

An essential characteristic of the algorithm comes precisely from the stochastic nature of the changes performed. The agent or ant moves to the worst adjacent vertex with a probability p_m (it moves randomly to any other adjacent vertex

with probability $1 - p_m$), and improves an edge connection, with probability p_c (otherwise it changes the edge at random). Both probabilities are adjustable parameters and allow the algorithm to escape from local minima and obtain close to the optimum. The process is repeated until a graph with the exact spectrum is found or the algorithm converges. The number of ants in the algorithm is another adjustable parameter that should increase with the diameter of the graph (the maximum of the distances between pairs of vertices). `Ants`

1. Generate an initial random graph. Put ants on it.
2. Repeat for each ant until stop criterion.
   (a) With probability p_m move ant to worst local vertex and change edges with probability p_c.
   (b) Evaluate *global* cost function.
   (c) Keep graph if better.

Cost function.

We have considered different cost functions to measure the quality of the solutions. All of them are based on the quadratic difference between the eigenvalues of the Laplacian of the tested and the reference graphs, $\sum_{i=0}^{n}(\lambda_i^0 - \lambda_i^t)^2$, and in some cases they include rank ponderation.

The cost functions tested are: $\sqrt{\sum_{i=0}^{n}(\lambda_i^0 - \lambda_i^t)^2}$, $\sqrt{\sum_{i=0}^{n}(i+1)(\lambda_i^0 - \lambda_i^t)^2}$, $\sqrt{\sum_{i=0}^{n}(n-i)(\lambda_i^0 - \lambda_i^t)^2}$, $\sqrt{\sum_{i=0}^{n}|\lambda_i^0 - \lambda_i^t|^3}$, and $\sqrt{\sum_{i=0}^{n}|\lambda_i^0 - \lambda_i^t|(n-1)^{1.5}}$.

The experiments with these cost functions to reconstruct random, regular, small-world, scale-free, and clustered graphs show that the quality of the reconstructions is comparable for all of the cost functions and graphs with a variation of at most ten percent in the worst case. The tests were performed considering for one hundred instances of each category of graph and using simulated annealing. The best results were obtained with the first simpler cost function, which was then used in main set of experiments. We have also tested sets of eigenvalues coming from the standard Laplacian, the normalized Laplacian, and the sign-less Laplacian and the coefficients of the characteristic polynomial. Again there are no important differences in the results obtained. Therefore, the final experiments were performed using the cost function $\sqrt{\sum_{i=0}^{n}(\lambda_i^0 - \lambda_i^t)^2}$ where the eigenvalues are those of the Laplacian matrix of the graph.

4 Results

The algorithms were implemented in C++ using Dev-C++ and executed on a PC (Pentium IV CPU at 2.41 GHZ) under Windows XP. Each reconstruction was limited to 300 seconds.

We tested the different algorithms and cost functions on graphs of small orders (up to 14 vertices) and in all cases we were able to reconstruct exactly the graph. We also performed experiments to check the behavior of the algorithm in the case of cospectral graphs. In particular, we ran 1000 reconstructions from

the spectrum $\{0, 3-\sqrt{5}, 2, 3, 3, 3+\sqrt{5}\}$ of the two coespectral graphs of order six of Figure 1, using the tabu search method and we obtained the graph on the left 554 times and the graph on the right 446 times.

Fig. 1. Two distinct Laplacian cospectral graphs of order six.

When reconstructing a network from a given spectrum, an identical (isomorphic) network with a different labeling of the vertices can result. To know if two networks are isomorphic is a difficult problem. It has been proved to belong to the class NP but it is thought not to be an NP-complete problem (see [11].) There is no known efficient (polynomial time) algorithm to solve this problem. Schmidt and Druffel [16] propose a method which we have implemented in our study. Two isomorphic networks should have the same exact degree distribution. After checking this property, we apply the Schmidt and Druffel algorithm which checks for isomorphism and returns a mapping between the vertices if they are isomorphic.

As stated in Section 2, a problem occurs in the reconstruction process when we want to compare two networks which are different but similar in some context (in our case their spectra are close). In this case, we consider the singular values decomposition method for their adjacency matrices and use the spectral distance defined there.

The combinatorial optimization algorithms and cost functions considered in our study were tested systematically as follows:

We generate one sample graph of order 40 for each of the categories considered: random, regular (circulant), Watts-Strogatz small-world [18], scale-free [3], and clustered. Figure 2 shows a graphic representation of the adjacency matrices of these graphs.

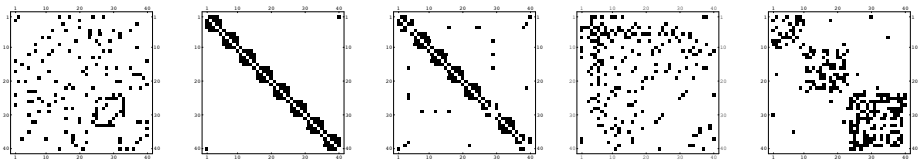

Fig. 2. Adjacency matrices of the reference graphs: random, regular (circulant), Watts-Strogatz small-world, scale-free and clustered. All graphs haver order 40.

For each reference graph, we compute the spectrum and use it to reconstruct the graph with the different optimization algorithms. To be fair in the comparisons among methods, we fix the reconstruction time for each graph to be 300 seconds. After this time, we compute the main topological parameters (diameter, average distance, degree distribution, clustering) for the best graph obtained and we check the similarity of its adjacency matrix with the original graph. Each test is repeated 100 times and the results are averaged.

Figure 3 shows a typical reconstruction: the left matrix corresponds to the adjacency matrix of the original graph (a clustered graph), the next matrix is the adjacency matrix of the reconstructed graph and the right matrix is the F matrix obtained from the this adjacency matrix after performing the singular value transformation. We see, in a visual way, the quality of the reconstruction (which can also be described more precisely by the parameters in Table 1.) In this figure, the matrix elements are represented by using gray-scale color maps whose limits are determined by their minimum and maximum values. Even though the matrix F does not coincide with the adjacency matrix of the reference graph, it is very close to it. Its distance (norm) to the reference graph matrix is $\delta = 0.07$ (see Section 2). In Table 1, we present a set of results for the tabu search method. More details for the whole set of experiments (other spectral distances) are available from the authors and are included in [10].

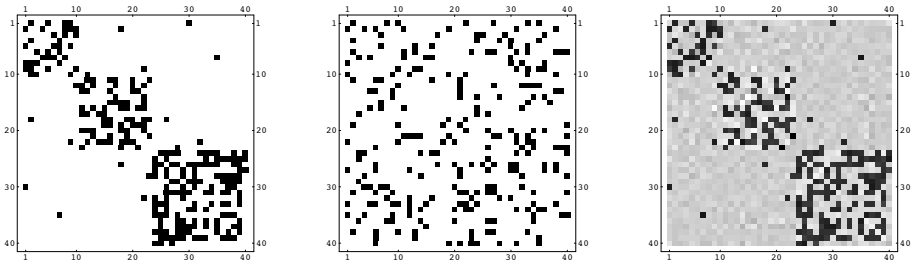

Fig. 3. Reconstruction of a clustered graph using tabu search. Left: The adjacency matrix of the original graph. Center: The adjacency matrix of the reconstructed graph. Right: Matrix F of the reconstructed graph.

Finally, in Table 2 we compute the values of the norm δ for the different combinatorial optimization methods. We note the these values are comparable with those in [14] where a simulated annealing algorithm with an elaborate cost function was used. Tests with a standard greedy algorithm [10] provide values that are around ten times worse for the same graphs.

Table 1. Tabu search. Results for the average of 100 reconstructions for each reference graph. Processing time: 300 seconds. Spectrum size: 40. Eigenvalue tolerance: 0.0001. Tabu list: 400. Tabu iterations after a change: 20.

		Random		Circulant		Small-world		Scale-free		Clustered	
		Orig.	*Rec.*	Orig.	*Rec.*	Orig.	*Rec.*	Orig.	*Rec.*	Orig.	*Rec.*
Diameter	min.	6	6	12	6	9	7	8	6	5	4
	avg.		6.4		9.7		7.9		6.6		5
	max.		7		18		9		8		6
Average distance llll	min.	2.894	2.822	6.185	1.032	3.973	3.451	3.057	2.824	2.646	2.174
	avg.		2.9		4.222		3.721		2.9032		2.226
	max.		2.964		7.441		3.956		3.046		2.277
Degree	min.	1	1	4	1	2	1	1	1	2	1
	avg.	3.8	3.8	4	4	3.9	3.9	3.45	3.45	6.3	6.3
	max.	8	8	4	5	6	6	12	12	13	13
Clustering	min.	0.199	0.163	0.7	0.604	0.529	0.461	0.446	0.468	0.372	0.189
	avg.		0.217		0.727		0.489		0.503		0.216
	max.		0.284		0.798		0.504		0.567		0.254
δ	min.		0.011		0.035		0.022		0.012		0.064
	avg.		0.017		0.079		0.042		0.019		0.076
	max.		0.050		0.145		0.064		0.034		0.088

Table 2. Results for the average of 100 reconstructions for each reference graph and combinatorial optimization method. The processing time is 300 seconds for reconstruction. The table shows values of the norm δ which measures similarity between a graph and its reconstruction from the Laplacian spectrum. The norm has values between 0 and 1 (see Section 2).

		Simulated annealing	Tabu search	Multiagent algorithm
Random	min.	0.011	0.011	0.014
	avg.	0.017	0.017	0.018
	max.	0.039	0.050	0.026
Circulant	min.	0.067	0.035	0.093
	avg.	0.090	0.079	0.098
	max.	0.099	0.145	0.104
Small-world	min.	0.037	0.022	0.053
	avg.	0.054	0.042	0.059
	max.	0.066	0.064	0.065
Scale-free	min.	0.011	0.012	0.150
	avg.	0.019	0.019	0.171
	max.	0.030	0.034	0.188
Clustered	min.	0.067	0.064	0.064
	avg.	0.077	0.076	0.079
	max.	0.089	0.088	0.085

5 Conclusion

The results show that all the three methods considered - simulated annealing, tabu search, and the multiagent algorithm *ants* - reconstruct small graphs exactly from their spectra and obtain a topologically good approximation for larger graphs. (We have tested graphs with up to 2000 nodes and 20000 edges.) Tabu search outperforms the other methods and we recommend it for its simplicity.

Our extensive tests also show that the best cost function is the quadratic difference of eigenvalues. It is much simpler than the spectral density cost function used in [14], is easier to implement, and results in faster algorithms.

References

1. E. Aarts, K. Lenstra, (Eds.). Local Search in Combinatorial Optimization. John Whiley & Sons Ltd. 1997.
2. J. Abril, F. Comellas, A. Cortés, J. Ozón, and M. Vaquer. A multi-agent system for frequency assignment in cellular radio networks. *IEEE Trans. Vehic. Technology* **49** (2000) 1558–1565.
3. A.-L. Barabasi, E. Bonabeau. Scale-free networks. *Scientific American* **288** No. 5 (2003) 50–59 .
4. M. Barahona and L. M. Pecora. Synchronization in small-world systems, *Phys. Rev. Lett.* **89** (2002) 054101.
5. F.R.K. Chung. Spectral Graph Theory. University of Pennsylvania, Philadelphia, PA, 1997.
6. F. Comellas, S. Gago. Synchronizability of complex networks. *J. Phys. A: Math. Theor.* **40** (2007) 4483–4492.
7. F. Comellas, E. Sapena. A multiagent algorithm for graph partitioning. *Lect. Notes in Comput. Sci.* **3907** (2006) 279–285.
8. D.M Cvetkovic, M. Doob, H. Sachs. Spectra of graphs. Theory and application. New York: Academic Press. 1980.
9. E.R. van Dam, W.H. Haemers. Which graphs are determined by their spectrum?. *Linear Algebra Appl.* **373** (2003) 241–272.
10. J. Diaz-Lopez. Reconstrucció espectral de grafs. *Projecte Final de Carrera* (Master's Thesis), January 2005. (In Catalan.)
11. M.R. Garey and D.S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York 1979.
12. F. Glover. Future paths for integer programming and links to artificial intelligence. *Comput. & Ops. Res.* **13** (1986) 533–549.
13. W.H. Haemers, E. Spence. Enumeration of cospectral graphs. *European J. Combin.* **25** (2004) 199–211.
14. M. Ipsen, A.S. Mikhailov. Evolutionary reconstruction of networks. *Phys. Rev. E.* **66** (2002) 046109.
15. S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi. Optimization by simulated annealing. *Science* **220** (1983) 671–680.
16. D.C. Schmidt, L.E. Druffel. A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices. *Journal of the ACM 23* (1976) 433–445.
17. G. Strang. Introduction to Linear Algebra. 3rd ed., Wellesley-Cambridge Press. 2003.
18. D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, *Nature* **393** (1998) 440–442.