Box-covering algorithm for fractal dimension of complex networks

Christian M. Schneider,[1,2,*] Tobias A. Kesselring,[1] José S. Andrade Jr.,[3] and Hans J. Herrmann[1,3]

[1]*Computational Physics, Institute for Building Materials, Eidgenössische Technische Hochschule Zürich, Schafmattstrasse 6, 8093 Zurich, Switzerland*

[2]*Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*

[3]*Departamento de Física, Universidade Federal do Ceará, 60451-970 Fortaleza, Ceará, Brazil*

(Received 6 April 2012; published 18 July 2012)

The self-similarity of complex networks is typically investigated through computational algorithms, the primary task of which is to cover the structure with a minimal number of boxes. Here we introduce a box-covering algorithm that outperforms previous ones in most cases. For the two benchmark cases tested, namely, the *E. coli* and the World Wide Web (WWW) networks, our results show that the improvement can be rather substantial, reaching up to 15% in the case of the WWW network.

DOI: 10.1103/PhysRevE.86.016707 PACS number(s): 05.10.−a, 64.60.aq, 89.75.Da, 89.75.Fb

I. INTRODUCTION

The topological and dynamical aspects of complex networks have been the focus of intensive research in recent years [1–15]. An open and unsolved problem in network and computer science is the question of how to cover a network with the fewest possible number of boxes of a given size [16–21]. In a complex network, a box size can be defined in terms of the chemical distance l_B, which corresponds to the number of edges on the shortest path between two nodes. This means that every node is less than l_B edges away from another node in the same box. Here we use the burning approach for the box-covering problem [22], thus the boxes are defined for a central node or edge. Instead of calculating the distance between every pair of nodes in a box, the maximal distance to the central node or edge r_B is given. While a solution for a given r_B is automatically a solution for a given l_B with the relation $r_B = (l_B - 1)/2$ for a central node and $r_B = l_B/2$ for a central edge, the solution for a given l_B is not necessarily a solution for its corresponding r_B. Therefore, the criterion of the distance between any two nodes is weaker and could result in a smaller number of boxes. Additionally, the usage of the distance l_B leads usually to disconnected boxes, while the boxes are always compact for r_B. The maximal chemical distance within a box of a given size r_B is $2r_B$ for a central node and $2r_B - 1$ for a central edge. Although this problem can be simply stated, its solution is known to be NP-hard [23]. It can also be mapped onto a graph coloring problem in computer science [19] and has important applications, e.g., the calculation of fractal dimensions of complex networks [24–29]. Here we introduce an efficient algorithm for fractal networks that is capable of determining the minimum number of boxes for the box-covering problem using the definition of a central node or edge for any given parameter r_B. Moreover, we compare it for two benchmark networks with a standard algorithm used to approximately obtain the minimal number of boxes.

In principle, the optimal solution should be identified by testing exhaustively all possible solutions. Nevertheless, for practical purposes, this approach is unfeasible since the solution space with its 2^N solutions is too large. Present algorithms such as maximum-excluded-mass-burning [22] and merging algorithms [30] are based on the sequential addition of the box with the highest score, e.g., the score is proportional to the number of covered nodes, and the boxes with the highest score are sequentially included. Other algorithms are based on simulated annealing [31], but without the guarantee of finding the optimal solution. Even greedy algorithms end up with approximately the same number of boxes as the algorithms mentioned before [20]. The greedy algorithm sequentially includes a node to a present box if all other nodes in this box are within the chemical distance l_B; if there is no such box, a new box with the new node is created. It is therefore believed that the results are close to the optimal result, although the real optimal solution is unknown.

This paper is organized as follows. In Sec. II, we introduce the algorithm and then explain the main difference between the present state of the art algorithm and our algorithm for a given distance r_B. In Sec. III, results for two benchmark networks are presented and the improvement in performance of our algorithm is quantitatively shown. Finally, in Sec. IV, we present conclusions and perspectives for future work.

II. ALGORITHM

We use two slightly different algorithms for the calculation of the box-covering solution, one for odd values of l_B and another for even values l_B. To get the results for an odd value, the following rules are applied.

(1) *Create all possible boxes*. For every node i create a box B_i containing all nodes that are at most $r_B = (l_B - 1)/2$ edges away. Node i is called the center of the box. An example is shown in Fig. 1(a).

(2) *Remove unnecessary boxes*. Search and remove all boxes B_i that are fully contained in another box B_j [see Fig. 1(b)].

(3) *Remove unnecessary nodes*. For every node i check all the boxes containing i: $B_{i_1}, \ldots, B_{i_n}$. If another node $j \neq i$ is contained in all of these boxes, remove it from *all* boxes [see Fig. 1(c)].

*schnechr@mit.edu

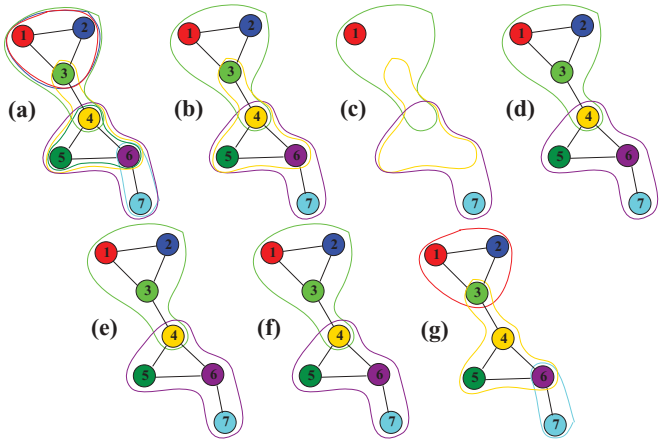

FIG. 1. (Color online) Box-covering algorithm on a small example network for the box size $l_B = 3$ ($r_B = 1$ with a central node). Top row: (a) Step 1: Calculation of all possible boxes. The color of the boxes corresponds to the node in its center. (b) Step 2: All boxes that are fully contained in another box are removed. In this example the boxes B_1, B_2, B_5, and B_7 are removed. (c) Step 3: All nodes that are in all boxes of another node are removed. In this example, nodes 2–4 are in the same box with node 1 and nodes 4–6 are in the same box with node 7. (d) The final solution is shown on the right side. Bottom row: The three possible solutions for the greedy box-covering algorithm, based on the largest box sizes. In this case, the boxes are included in the solution according to the number of new covered nodes. Since three boxes B_3, B_4, and B_6 have the same number of nodes, the algorithm finds three different solutions: (e) (B_3,B_6), (f) (B_6,B_3), and (g) (B_4,B_1,B_7), where the last one is not optimal.

(4) *Remove pairs of unnecessary twin boxes*. Find two nodes i and j that are both in exactly two boxes of size 2: $B_{i_1} = \{i,k_1\}$, $B_{i_2} = \{i,k_2\}$ and $B_{j_1} = \{j,l_1\}$, $B_{j_2} = \{j,l_2\}$. If $k_1 = l_1$ and $k_2 = l_2$, then B_{i_2} and B_{j_1} can be removed. If $k_1 = l_2$ and $k_2 = l_1$, then B_{i_2} and B_{j_2} can be removed. An example for this rule is shown in Fig. 2. Note that such twin boxes also appear for $l_B > 2$ due to the removal of unnecessary nodes.

(5) *Search for boxes that must be contained in the solution*. Add all boxes B_i to the solution that have a node i only present in this box. Remove all nodes $j \neq i$ covered by B_i from other boxes.

(6) *Iterate A*. Repeat steps 2–5 until there is no node that is covered by a single box and is not part of the solution.

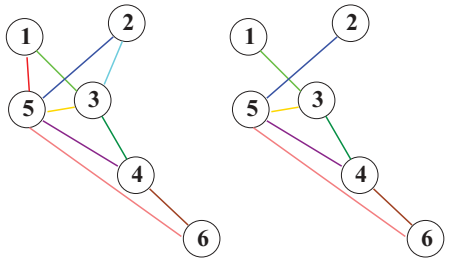

FIG. 2. (Color online) Step 4: In this example two nodes are in the same box if they are connected with an edge. The two boxes between nodes 1 and 5 and between nodes 2 and 3 are removed according to rule 4.

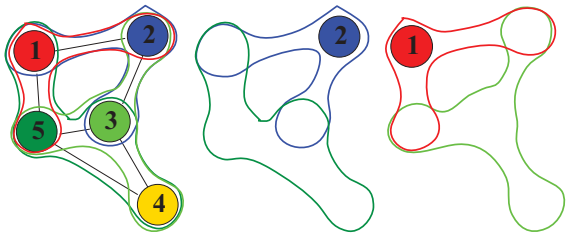

FIG. 3. (Color online) Step 8: Node 4 is covered by two circles (the minimal number of boxes) and the algorithm splits. The first subalgorithm continues with box B_5 (middle), while the second one continues with box B_3 (right).

(7) *System split*. Identify if the remaining network can be divided into subnetworks such that all boxes in a subnetwork contain only nodes of this subnetwork. Then these subnetworks can be processed independently from each other.

(8) *System split*. Find the node that is in the smallest number of boxes N_{boxes}; each of these boxes covers another set of nodes B_i. If there is more than one node fulfilling this criterion, choose the node that is covered by the largest boxes. Then the algorithm is divided into N_{boxes} subalgorithms, which can be independently calculated in parallel. By removing from each of the N_{boxes} subalgorithm another set of nodes B_i, all possible solutions are considered. An example for the splitting is shown in Fig. 3. Since we want to identify only the best solution, we do not need to calculate the results of all subalgorithms. As soon as one of the subalgorithms identifies the best possible solution (the number of boxes included in the solution of the subalgorithm is not larger than the rounded up number of uncovered nodes divided by the number of nodes in the largest box), we can skip the calculation of the others. Furthermore, the calculation of a subalgorithm can be skipped if the minimal number of required additional boxes reaches the number of the (so far) best solution of a parallel subalgorithm.

(9) *Iterate B*. Repeat steps 2–8 until no nodes are uncovered.

(10) *Identify the best solution*. Choose the solution with the lowest number of boxes. This is the best solution for a given r_B.

To get the results for an even value of l_B the first step is slightly different.

(1′) *Create all possible boxes*. For every edge i create a box B_i containing all nodes that are at most $r_B = l_B/2$ nodes away. Edge i is called the center of the box.

All other steps are the same as for the odd case. Note that the number of starting boxes for odd values scales with the number of nodes of the network N and with the number of edges M for even values. Due to the fact that the problem is NP-hard, the required computational time depends strongly on the network itself. For example, the box-covering for tree networks could be performed in $O(N^3)$, while for regular networks it requires $O(2^N)$.

III. RESULTS FOR TWO BENCHMARK NETWORKS

In the following, we will argue why the algorithm leads to a nearly optimal solution. Instead of sequentially including boxes, the idea of our algorithm is to remove all undesired

boxes from the solution space ending up with a final, best solution. To reduce the huge solution space, our box-covering algorithm uses two basic ingredients: (i) Unnecessary boxes from the solution space are discarded and the boxes that definitively belong to the solution are kept and (ii) unnecessary nodes from the network are discarded. These two steps reduce the solution space of a wide range of network types significantly, especially if they are applied in alternation as the removal of a box can lead to the removal of nodes and other boxes and vice versa. Nevertheless, these two steps do not necessarily lead to the optimal solution, thus the solution space has to be split into several possible subsolution spaces. In each of these subsolutions the first two steps are repeated. Note that the splitting does not reduce the number of possible solutions, thus only the first two steps reduce the solution space and in the worst case, the algorithm must calculate the entire solution space. In any case, for many complex networks iterating these three steps significantly reduces the solution space to a few solutions from which the best box covering can be obtained.

The remaining question is how to judge whether a box or node is necessary or unnecessary. On the one hand, a box is unnecessary if all nodes of a box are also part of another box. This box can be removed because the other box covers at least the same nodes and often additional nodes. On the other hand, a box is necessary if a node is exclusively covered by this single box. This box has to belong to the solution since only if the box is part of the solution, the node is covered.

In contrast, nodes can easily be identified as unnecessary. For example, all nodes of a box, which is part of the solution, can be removed from all other boxes since they are already covered. Additionally, if a node shares all boxes with another node, the other node can be removed since the second node is always covered if the first node is covered. These few rules are in principle sufficient to get the best solution since our algorithm starts with *all* 2^N or 2^M (for central edges) possible solutions and discards unnecessary and includes necessary boxes. Without any approximation for the classification of boxes, the algorithm ends up with the best number of boxes for any given radius r_B.

Although we calculate results for only undirected, unweighted networks, the algorithm can easily be extended to directed and weighted networks. In both cases only the initial step, the creation of boxes, is different. For directed networks, the box around a central node contains all nodes that are reachable with respect to the direction, while for weighted networks, the distance is the sum of the edge weights between the nodes.

Next we show that our algorithm can also identify solutions for large networks. Therefore, we have applied it to two different benchmark networks, namely, the *E. coli* network [32], with 2859 proteins and 6890 interactions between them, and the World Wide Web (WWW) network [2]. We compare the results for the minimal box number $N(l_B)$ of our algorithm for different values of box sizes l_B with the results of the greedy graph coloring algorithm [22], as displayed in Fig. 4. In contrast to our algorithm, the greedy algorithm is based on the distance between any pair of nodes within a box. While the absolute improvement is rather small, the relative improvement is up to 6% for $l_B < 7$. If the network is fractal, it should obey

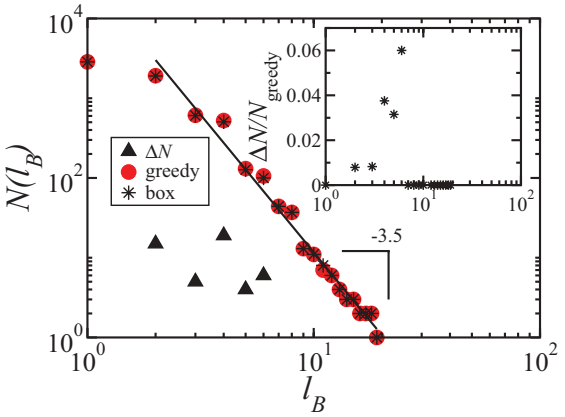

FIG. 4. (Color online) Comparison of the minimal number of boxes $N(l_B)$ for a given distance l_B for the *E. coli* network using the greedy graph coloring algorithm and our algorithm. While the decay for both box-covering methods is similar in the logarithmic plot, the minimal number of boxes is different. Although the difference $\Delta N = N_{\text{greedy}} - N_{\text{box}}$ seems to be small, the relative improvement $\Delta N/N_{\text{greedy}}$, which is shown in the inset, is significant for small distances $l_B < 7$. Note that the larger the box size the simpler the network can be covered with the adequate number of boxes. The straight line shows a power-law behavior, where the best fit for the fractal dimension is $d_B = 3.47 \pm 0.11$ for the greedy graph coloring and $d_B = 3.45 \pm 0.10$ for our algorithm, respectively. Within the error bars both box-covering algorithms yield the same fractal dimension.

the relation

$$N(l_B) \sim l_B^{-d_B}, \tag{1}$$

where d_B is the fractal dimension. Interestingly, it seems that the fractal dimension $d_B = 3.47 \pm 0.11$ from the greedy algorithm and $d_B = 3.45 \pm 0.10$ from our algorithm of the network is within the errors unaffected by the choice of the

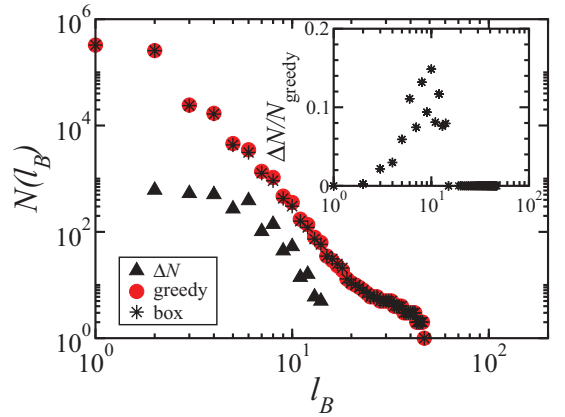

FIG. 5. (Color online) Minimal number of boxes $N(l_B)$ as a function of the distance l_B for the WWW network calculated through the greedy graph coloring algorithm and our algorithm. While the fractal dimension for both box-covering methods agrees within the error bars, the minimal number of boxes is different. The difference $\Delta N = N_{\text{greedy}} - N_{\text{box}}$, as well as the relative improvement $\Delta N/N_{\text{greedy}}$, which is shown in the inset, is significant for $l_B < 16$. For this network a maximal relative improvement of about 15% can be obtained.

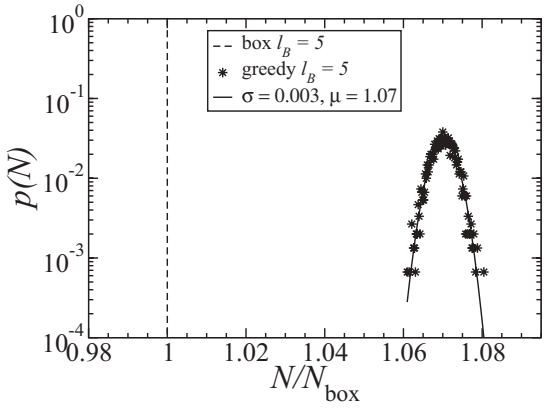

FIG. 6. Distribution of minimal number of boxes $p(N)$ for the WWW network for $l_B = 5$ calculated through the greedy graph coloring algorithm for 1500 different random node sequences. We have normalized the results by the solution obtained from our algorithm. The distribution follows a normal distribution $p(x) \sim \exp[-(x-\mu)^2/2\sigma^2]$ with $\mu = 1.07 \pm 0.01$ and $\sigma = 0.003 \pm 0.001$, thus approximately 10^{120} realizations are necessary to find the solution with the greedy algorithm.

algorithm and the definition of box-covering problem. Note that for $l_B = 11$, due to the fact that the boxes are calculated based on different box definitions, we have one more box. The simplest case where such a difference occurs is in a chain of four connecting nodes (1-2, 2-3, 3-4, and 4-1). All nodes have a chemical distance of 2 between them ($l_B = 3$); however, it is not possible to draw a box around a node with radius one ($r_B = 1$), which contains all nodes.

The second example is the WWW network, containing 325 729 nodes and 1 090 108 edges. As in the previous case, our algorithm outperforms the state of the art algorithm, but yields similar fractal behavior, as shown in Fig. 5. For intermediate box sizes $l_B < 16$, we have a large improvement since up to 15% and up to 611 fewer boxes are needed. For $l_B = 16, 17, 18$ we have two boxes more, like in the *E. coli* network case due to the two definitions of the box-covering problem, while for larger l_B both algorithms give similar results. Interestingly, it seems that the improvement for even distances l_B (for central edges) is significantly larger than for odd distances l_B (for central nodes).

In Fig. 6 we show the influence of the sequence of adding nodes to the boxes on the results of the greedy algorithm. While the results of Fig. 5 are the minimal values obtained from 50 independent starting sequences, we calculated 1500 realizations for a single box size $l_B = 5$. The difference between the improvement is, with $N_{\text{greedy}}/N_{\text{box}} = 6.3\%$ and 6.1%, rather small. The gap between our solution and the greedy algorithm is too large; thus, for practical purposes, the greedy algorithm will never find our solution for this box size.

The results for these two benchmark networks demonstrate that our algorithm is more effective than the state of the art algorithms. Nevertheless, due to the rapid decay of the number of boxes for larger box sizes, the fractal dimension of the two benchmark networks are the same within the standard errors when using our box-covering algorithm and other algorithms.

IV. CONCLUSION

In closing, we have presented a box-covering algorithm that identifies the least number of boxes for a box with a central node or edge. We have also compared our algorithm with the state of the art method for different benchmark networks and detected substantial improvements, although our method uses a stronger definition for boxes. The obtained solutions are nearly optimal as a consequence of the algorithm design if the box size is defined as the maximal distance r_B to the central node or edge. Moreover, we believe that our algorithm can identify one optimal solution for this definition and it would be an interesting challenge for the future to try to prove or disprove this hypothesis. Our approach can be useful for designing efficient commercial distribution networks, where the shops are the nodes, the storage facilities are the box centers, and the radius is related to the boundary conditions, such as transportation cost or time.

ACKNOWLEDGMENTS

We acknowledge financial support from the Eidgenössische Technische Hochschule Zürich (ETH) Competence Center "Coping with Crises in Complex Socio-Economic Systems" through ETH Research Grant No. CH1-01-08-2 and by the Swiss National Science Foundation under Contract No. 200021 126853. We also thank the Brazilian agencies Conselho Nacional de Desenvolvimento Científico e Tecnológico, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico, and the Instituto Nacional de Cîencia e Tecnologia para Sistemas Complexos (INST-SC) for financial support.

[1] D. Watts and S. Strogatz, Nature (London) **393**, 440 (1998).
[2] R. Albert, H. Jeong, and A.-L. Barabási, Nature (London) **401**, 130 (1999).
[3] M. Barthélémy and L. A. N. Amaral, Phys. Rev. Lett. **82**, 5180 (1999).
[4] A. L. Lloyd and R. M. May, Science **292**, 1316 (2001).
[5] R. Cohen, S. Havlin, and D. ben-Avraham, Phys. Rev. Lett. **91**, 247901 (2003).
[6] M. Barthélémy, A. Barrat, R. Pastor-Satorras, and A. Vespignani, Phys. Rev. Lett. **92**, 178701 (2004).
[7] M. C. González, P. G. Lind, and H. J. Herrmann, Phys. Rev. Lett. **96**, 088702 (2006).
[8] L. K. Gallos, C. Song, S. Havlin, and H. A. Makse, Proc. Natl. Acad. Sci. USA **104**, 7746 (2007).
[9] A. A. Moreira, J. S. Andrade Jr., H. J. Herrmann, and J. O. Indekeu, Phys. Rev. Lett. **102**, 018701 (2009).

[10] H. Hooyberghs, B. Van Schaeybroeck, A. A. Moreira, J. S. Andrade Jr., H. J. Herrmann, and J. O. Indekeu, Phys. Rev. E **81**, 011102 (2010).
[11] G. Li, S. D. S. Reis, A. A. Moreira, S. Havlin, H. E. Stanley, and J. S. Andrade Jr., Phys. Rev. Lett. **104**, 018701 (2010).
[12] H. J. Herrmann, C. M. Schneider, A. A. Moreira, J. S. Andrade Jr., and S. Havlin, J. Stat. Mech. (2011) P01027.
[13] C. M. Schneider, A. A. Moreira, J. S. Andrade Jr., S. Havlin, and H. J. Herrmann, Proc. Natl. Acad. Sci. USA **108**, 3838 (2011).
[14] C. M. Schneider, T. Mihaljev, S. Havlin, and H. J. Herrmann, Phys. Rev. E **84**, 061911 (2011).
[15] A. Vespignani, Nature Phys. **8**, 32 (2012).
[16] H. O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and Fractals: New Frontiers of Science* (Springer, Berlin, 1993).
[17] J. Feder, *Fractals* (Plenum, New York, 1988).
[18] *Fractals in Science*, edited by A. Bunde and S. Havlin (Springer, Berlin, 1995).
[19] *Graph Coloring Problems*, edited by T. R. Jensen and B. Toft (Wiley-Interscience, New York, 1995).
[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms* (MIT, Cambridge, 2001).
[21] C. Song, S. Havlin, and H. A. Makse, Nature (London) **433**, 392 (2005).
[22] C. Song, L. K. Gallos, S. Havlin, and H. A. Makse, J. Stat. Mech. (2007) P03006.
[23] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).
[24] S. H. Yook, F. Radicchi, and H. Meyer-Ortmanns, Phys. Rev. E **72**, 045105 (2005).
[25] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005).
[26] F. C. Zhao, H. J. Yang, and B. Wang, Phys. Rev. E **72**, 046119 (2005).
[27] C. Song, S. Havlin, and H. A. Makse, Nature Phys. **2**, 275 (2006).
[28] K.-I. Goh, G. Salvi, B. Kahng, and D. Kim, Phys. Rev. Lett. **96**, 018701 (2006).
[29] A. A. Moreira, D. R. Paula, R. N. Costa Filho, and J. S. Andrade Jr., Phys. Rev. E **73**, 065101 (2006).
[30] M. Locci, G. Concas, R. Tonelli, and I. Turna, WSEAS Trans. Inf. Sci. Appl. **7**, 371 (2010).
[31] W. X. Zhou, Z. Q. Jiang, and D. Sornette, Physica A **375**, 741 (2007).
[32] http://lev.ccny.cuny.edu/~hmakse/METHODS/methods.html.