How to calculate the fractal dimension of a complex network: the box covering algorithm

Chaoming Song[1], Lazaros K Gallos[1], Shlomo Havlin[2] and Hernán A Makse[1]

[1] Levich Institute and Physics Department, City College of New York, New York, NY 10031, USA
[2] Minerva Center and Department of Physics, Bar-Ilan University, 52900 Ramat-Gan, Israel
E-mail: chaoming_song@msn.com, gallos@physics.auth.gr, havlin@ophir.ph.biu.ac.il and hmakse@levdec.engr.ccny.cuny.edu

Abstract. Covering a network with the minimum possible number of boxes can reveal interesting features for the network structure, especially in terms of self-similar or fractal characteristics. Considerable attention has been recently devoted to this problem, with the finding that many real networks are self-similar fractals. Here we present, compare and study in detail a number of algorithms that we have used in previous papers towards this goal. We show that this problem can be mapped to the well-known graph colouring problem and then we simply can apply well-established algorithms. This seems to be the most efficient method, but we also present two other algorithms based on burning which provide a number of other benefits. We argue that the algorithms presented provide a solution close to optimal and that another algorithm that can significantly improve this result in an efficient way does not exist. We offer to anyone that finds such a method to cover his/her expenses for a one-week trip to our lab in New York (details in http://jamlab.org).

Keywords: analysis of algorithms, growth processes

J. Stat. Mech. (2007) P03006

Contents

1. Introduction

Complex networks are important since they describe efficiently many social, biological and communication systems [1]–[5]. There exist many kinds of networks and characterizing their topology is very important for a wide range of static and dynamic properties. Recently [6, 7], we applied a box covering algorithm which enabled us to demonstrate the existence of self-similarity in many real networks. The fractal and self-similarity properties of complex networks were subsequently studied extensively in a variety of systems [8]–[17]. In this paper we provide a detailed study of the algorithms used to calculate quantities characterizing the topology of such networks, such as the fractal dimension d_B. We study and compare several possible box covering algorithms, by applying them to a number of model and real-world networks and we relate the box covering optimization to the well-known vertex colouring algorithm [18]. We also suggest a new definition for the box size ℓ_B, which seems to yield more accurate values for the fractal dimension d_B of a complex network.

We show that the optimal network covering can be directly mapped to a vertex colouring problem, which is a well-studied problem in graph theory. Although we use a specific version of the greedy colouring algorithm it is possible that other colouring algorithms may be used. We find that this approach leads to the most efficient solution of the optimal box covering problem, but we also present two other methods based on breadth-first search which address certain disadvantages of the first method, such as disconnected or non-compact boxes.

We also compare our results with a number of methods introduced by others for studying this problem. For example, Kim *et al* [19] have used a variation of the random burning method, where a random node serves as the seed of a box and neighbouring unburned nodes are assigned to this box. A similar method was applied for edge covering (instead of node covering) which yields similar results [20].
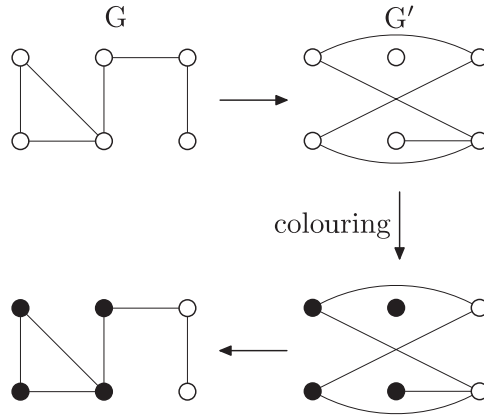
Figure 1. Illustration of the solution for the network covering problem via mapping to the graph colouring problem. Starting from G (upper left panel) we construct the dual network G' (upper right panel) for a given box size (here $\ell_B = 3$), where two nodes are connected if they are at a distance $\ell \geq \ell_B$. We use a greedy algorithm for vertex colouring in G', which is then used to determine the box covering in G, as shown in the plot.

2. The greedy colouring algorithm

We begin by recalling the original definition of box covering by Hausdorff [21]–[23]. For a given network G and box size ℓ_B, a box is a set of nodes where all distances ℓ_{ij} between any two nodes i and j in the box are smaller than ℓ_B. The minimum number of boxes required to cover the entire network G is denoted by N_B. For $\ell_B = 1$, N_B is obviously equal to the size of the network N, while $N_B = 1$ for $\ell_B \geq \ell_B^{max}$, where ℓ_B^{max} is the diameter of the network (i.e. the maximum distance in the network) plus one.

The ultimate goal of all box covering algorithms is to locate the optimum solution, i.e., to identify the minimum $N_B(\ell_B)$ value for any given box size ℓ_B. We first demonstrate that this problem can be mapped to the graph colouring problem, which is known to belong to the family of NP-hard problems [24]. This means that an algorithm that can provide an exact solution in a relatively short amount of time does not exist. This concept, though, enables us to treat the box covering problem using known optimization approximations. In order to find an approximation for the optimal solution for an arbitrary value of ℓ_B we first construct a dual network G', in which two nodes are connected if the chemical distance between them in G (the original network) is greater than or equal to ℓ_B. In figure 1 we demonstrate an example of a network G which yields such a dual network G' for $\ell_B = 3$ (upper row of the figure).

Vertex colouring is a well-known procedure, where labels (or colours) are assigned to each vertex of a network, so that no edge connects two identically coloured vertices. It is clear that such a colouring in G' gives rise to a natural box covering in the original network G, in the sense that vertices of the same colour will necessarily form a box since the distance between them must be less than ℓ_B. Accordingly, the minimum number of boxes $N_B(G)$ is equal to the minimum required number of colours (or the chromatic number) in the dual network G', $\chi(G')$, which is a famous problem in traditional graph theory.

In simpler terms, (a) if the distance between two nodes in G is greater than ℓ_B these two neighbours cannot belong in the same box. According to the construction of G', these two nodes will be connected in G' and thus they cannot have the same colour. Since they have a different colour they will not belong in the same box in G, which is our initial assumption. (b) On the contrary, if the distance between two nodes in G is less than ℓ_B it is possible that these nodes belong in the same box. In G' these two nodes will not be connected and it is allowed for these two nodes to carry the same colour, i.e. they may belong to the same box in G, (whether these nodes will actually be connected depends on the exact implementation of the colouring algorithm, to be discussed later).

The exact solution for vertex colouring can only be achieved on small size networks, since the optimal number of colours in an arbitrary graph is an NP-hard problem, as mentioned above, and in general should be solved by a brute-force approach [25, 26]. In practice, a greedy algorithm is widely adopted to obtain an approximate solution [27] and this also works very well for our case of box covering. We implement a simple version of the greedy algorithm as follows: (1) Rank the nodes in a sequence, (2) Mark each node with a free colour, which is different from the colours of its nearest neighbours in G'. The algorithm that follows both constructs the dual network G' and assigns the proper node colours for all ℓ_B values in one pass. For this implementation we need a two-dimensional matrix $c_{i\ell}$ of size $N \times \ell_B^{\max}$, whose values represent the colour of node i for a given box size $\ell = \ell_B$.

(i) Assign a unique id from 1 to N to all network nodes, without assigning any colours yet.

(ii) For all ℓ_B values, assign a colour value 0 to the node with id $= 1$, i.e. $c_{1\ell} = 0$.

(iii) Set the id value $i = 2$. Repeat the following until $i = N$.

    (a) Calculate the distance ℓ_{ij} from i to all the nodes in the network with id j less than i.

    (b) Set $\ell_B = 1$.

    (c) Select one of the unused colours $c_{j\ell_{ij}}$ from all nodes $j < i$ for which $\ell_{ij} \geq \ell_B$. This is the colour $c_{i\ell_B}$ of node i for the given ℓ_B value.

    (d) Increase ℓ_B by one and repeat (c) until $\ell_B = \ell_B^{\max}$.

    (e) Increase i by 1.

This greedy algorithm is very efficient, since we can cover the network with a sequence of box sizes ℓ_B performing only one network pass.

The results of the greedy algorithm may depend on the original colouring sequence. In order to investigate the quality of the algorithm, we randomly reshuffle the colouring sequence and apply the greedy algorithm for 10 000 times on several different models and real-world networks. In figure 2 we present a typical example for the PDFs of N_B for the cellular network of *E. coli*. The curves for all box sizes ℓ_B are narrow Gaussian distributions, indicating that almost any implementation of the algorithm yields a solution close to the optimal.

The uncertainty of the algorithm can be quantified via the normalized variances $\sigma_B \equiv \langle \Delta N_B^2 \rangle^{1/2} / \langle N_B \rangle$ of the PDFs. In figure 3, we present the σ_B dependence on the box size ℓ_B for both fractal (left panel) and non-fractal (right panel) networks. Surprisingly,
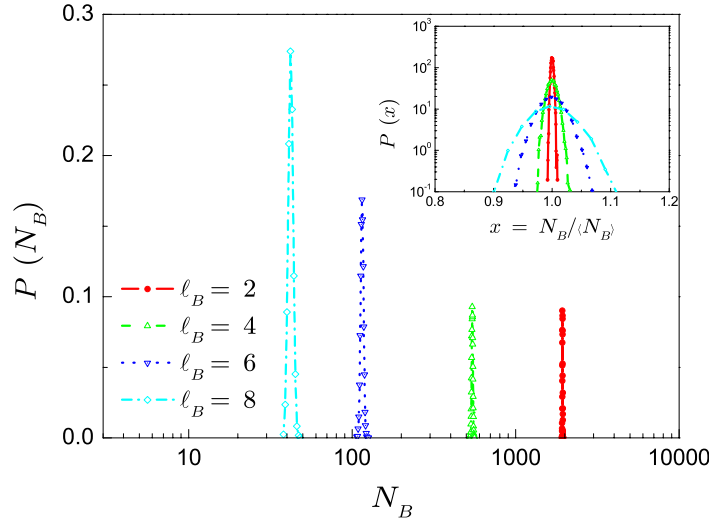
Figure 2. Probability distribution function $P(N_B)$ of the number of boxes N_B for the greedy algorithm, applied to the cellular network of *E. coli*. Different box sizes ℓ_B are used as indicated in the plot. Inset: PDFs of the normalized quantity $N_B/\langle N_B \rangle$ in a semi-log plot for the greedy algorithm, suggesting that $P(N_B)$ follows a Gaussian distribution.
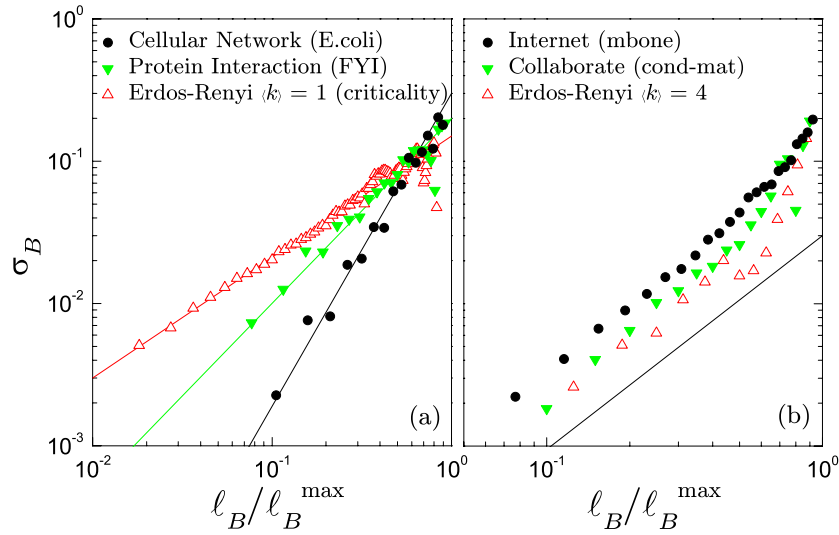

Figure 3. Normalized variance σ_B of the greedy algorithm for different box sizes ℓ_B, for (a) fractal and (b) non-fractal networks, where the box size ℓ_B is normalized by the maximum box size $\ell_B^{\max}$. The slopes for the fractal networks are (left to right): $\delta = 0.85, 1.3, 2.2$. For non-fractal networks: $\delta = 1.5$.

when $\ell_B \ll \ell_{\max}$ all the networks seem to exhibit a power-law dependence

$$\sigma_B \sim \ell_B^{\delta}, \tag{1}$$

even for the case of non-fractal networks. In fractal networks the value of δ depends on the network structure, while for non-fractal networks δ seems to be constant with a value close to 1.5.
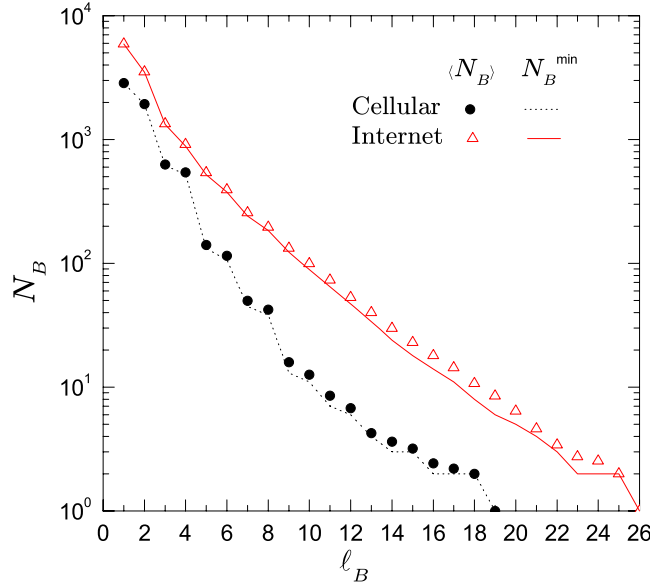
Figure 4. Comparison of the minimum $N_{\mathrm{B}}^{\mathrm{min}}$ (line) and mean $\langle N_{\mathrm{B}} \rangle$ (symbols) number of boxes for the greedy algorithm after $10\,000$ random reshuffles in real-world networks.

Strictly speaking, the calculation of the fractal dimension d_{B} through the relation $N_{\mathrm{B}} \sim \ell_{\mathrm{B}}^{-d_{\mathrm{B}}}$ is valid only for the minimum possible value of N_{B}, for any given ℓ_{B} value, so an algorithm should aim to find this minimum N_{B}. For the greedy colouring algorithm it has been shown [27] that it can identify a colouring sequence which yields the optimal solution, i.e. the minimal value from the greedy algorithm coincides with the optimal value. Obviously, there is no rule as to when this minimum value has been actually reached. Yet, it is still meaningful to compare the mean value $\langle N_{\mathrm{B}} \rangle$ with the minimum value $N_{\mathrm{B}}^{\mathrm{min}}$ for our sample of $10\,000$ different realizations. We present such a comparison for the cellular network (fractal) and the Internet (non-fractal) in figure 4. For all ℓ_{B} values the difference between $\langle N_{\mathrm{B}} \rangle$ and $N_{\mathrm{B}}^{\mathrm{min}}$ is very small and the two values are almost indistinguishable from each other. This result is significant for implementation purposes, by pointing out that any realization of the above algorithm practically yields a quite accurate outcome.

The presented greedy algorithm is one of the simplest algorithms capable to solve the exact colouring problem. The colouring problem is very important in many fields, though, and consequently there is an enormous amount of studies on this subject. In principle, any one of the suggested algorithmic solutions in the literature can also be adopted for dealing with the box covering problem.

The form of the algorithm that was described above is the one that was used in [6,7] for the calculation of N_{B} versus ℓ_{B}.

3. Burning algorithms

The presented greedy colouring algorithm provides at the same time high efficiency and significant accuracy. A simpler approach, though, is to use more traditional breadth-first algorithms. In the following sections we describe the basic simple burning algorithm and
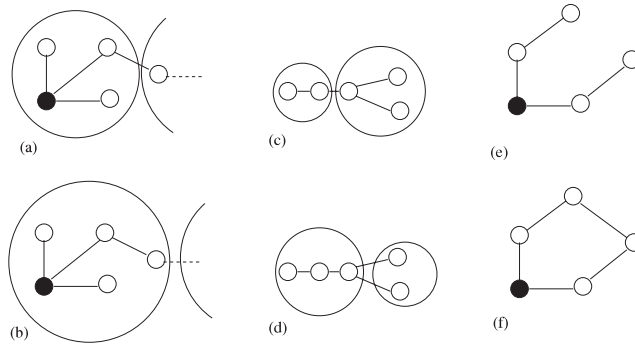
Figure 5. Our definitions for a box that is (a) non-compact for $\ell_B = 3$, i.e. could include more nodes, (b) compact, (c) connected, and (d) disconnected (the nodes in the right box are not connected in the box). (e) For this box, the values $\ell_B = 5$ and $r_B = 2$ verify the relation $\ell_B = 2r_B + 1$. (f) One of the pathological cases where this relation is not valid, since $\ell_B = 3$ and $r_B = 2$.

introduce two alternative (more sophisticated) methods based on similar ideas. We then proceed to compare these algorithms to the greedy colouring algorithm.

In the following, we define a box to be 'compact' when it includes the maximum possible number of nodes, i.e. when there do not exist any other network nodes that could be included in this box. A 'connected' box means that any node in the box can be reached from any other node in this box, without having to leave this box. Equivalently, a 'disconnected' box denotes a box where certain nodes can be reached by other nodes in the box only by visiting nodes outside this box. For a demonstration of these definitions see figure 5.

A short note on the definition of the distances used. A box of size ℓ_B, according to our definition, includes nodes where the distance between any pair of nodes is less than ℓ_B. It is possible, though, to grow a box from a given central node, so that all nodes in the box are within distance less than a given box radius r_B (the maximum distance from a central node). For the original definition of the box, ℓ_B corresponds to the box diameter (maximum distance between any two nodes in the box) plus one. Thus, these two measures are related through $\ell_B = 2r_B + 1$. In general this relation is valid for random configurations, but there may exist specific cases, such as e.g. nodes in a cycle, where this equation is not exact (figure 5).

3.1. Burning with the diameter ℓ_B, and the compact-box-burning (CBB) algorithm

A traditional geometrical approach is the so-called 'burning' algorithm (breadth-first search). The basic idea is to generate a box by growing it from one randomly selected node towards its neighbourhood until the box is compact, or equivalently that each box should include the maximum possible number of nodes. The algorithm is quite simple and can be summarized as follows:

(i) Choose a random uncovered node as the seed for a new box.

(ii) All uncovered nodes connected to the current box are tested for being within distance ℓ_B from all the nodes currently in the box. Nodes that obey this criterion are included in the box.
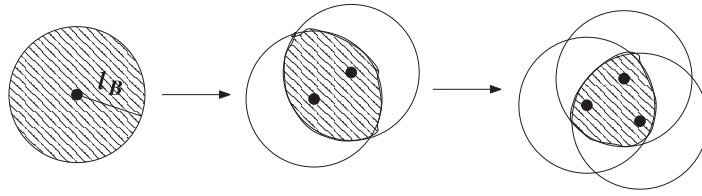
Figure 6. Two-dimensional geometrical analogue of the CBB algorithm. Initially we choose a random point and consider the circle with radius ℓ_B. We then choose another random point within this circle which serves as a new circle centre and calculate the union of these two circles. We continue by iteratively selecting random centres for circles in the union of all the previous circles.

(iii) Repeat (ii) until there are no more nodes that can be added in this box.

(iv) Repeat (i), (ii) and (iii) until all nodes are covered.

Although this algorithm is quite easy to implement, it requires a very long computational time. For this reason, we introduce a method that yields the exact same results as the above algorithm, but is computationally less intensive and can be executed much faster. We call this algorithm the compact-box-burning or CBB algorithm.

The method can be better understood in geometrical terms (figure 6). We start from a random point and draw a circle with radius ℓ_B. We then select a random point within this circle and draw a circle with radius ℓ_B using this new centre. The union of the two circles includes all possible points that will eventually form the box. Iteratively adding points from the union of all previous circles and drawing new circles we eventually create a box where all the included points are within distance ℓ_B from each other. For the case of a complex network, we apply the following algorithm (see figure 7):

 (i) Construct the set C of all yet uncovered nodes.

 (ii) Choose a random node p from the candidate set C and remove it from C.

(iii) Remove from C all nodes i whose distance from p is $\ell_{pi} \geq \ell_\mathrm{B}$, since by definition they will not belong in the same box.

(iv) Repeat steps (ii) and (iii) until the candidate set is empty.

The set of the chosen nodes p forms a compact box. We then repeat the above procedure until the entire network is covered.

We also performed 10 000 realizations for the CBB algorithm and calculated the mean value $\langle N_\mathrm{B} \rangle$ and the normalized variance σ_B. In figure 8 we compare the greedy algorithm with CBB for both fractal and non-fractal networks. The value of $\langle N_\mathrm{B} \rangle$ is roughly the same for both algorithms, with the value from CBB slightly larger (at most 2%) than the one from the greedy algorithm. More interestingly, the normalized variances are very close for these two algorithms. This suggests that CBB provides results comparable with the greedy algorithm, but CBB may be a bit simpler to implement.

3.2. Burning with the radius r_B, and the maximum-excluded-mass-burning (MEMB) algorithm

The formal definition of boxes includes the maximum separation ℓ_B between any two nodes in a box. However, it is possible to recover the same fractal properties of a network,

● Belongs to box
● Just selected
◎ Candidate
○ Out of box
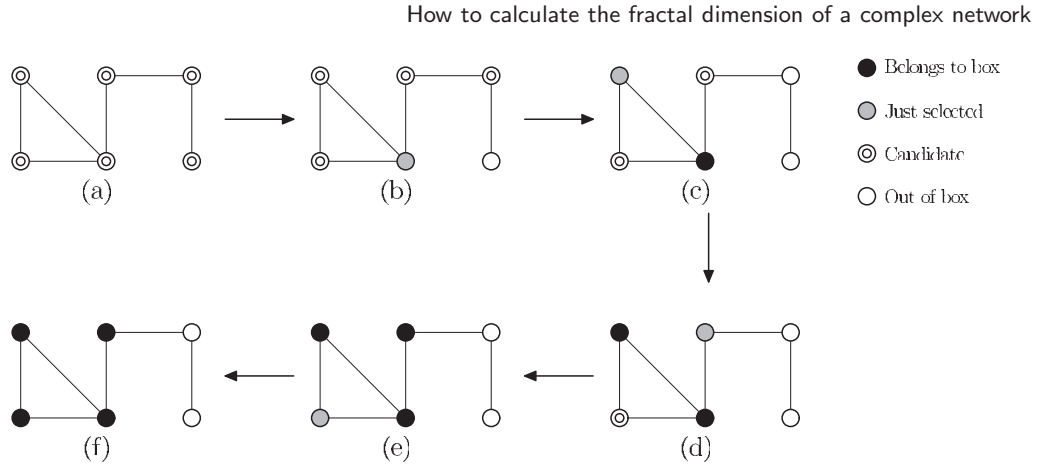
Figure 7. Illustration of the CBB algorithm for $\ell_B = 3$. (a) Initially, all nodes are candidates for the box. (b) A random node is chosen, and nodes at a distance further than ℓ_B from this node are no longer candidates. (c) The node chosen in (b) becomes part of the box and another candidate node is chosen. The above process is then repeated until the box is complete.
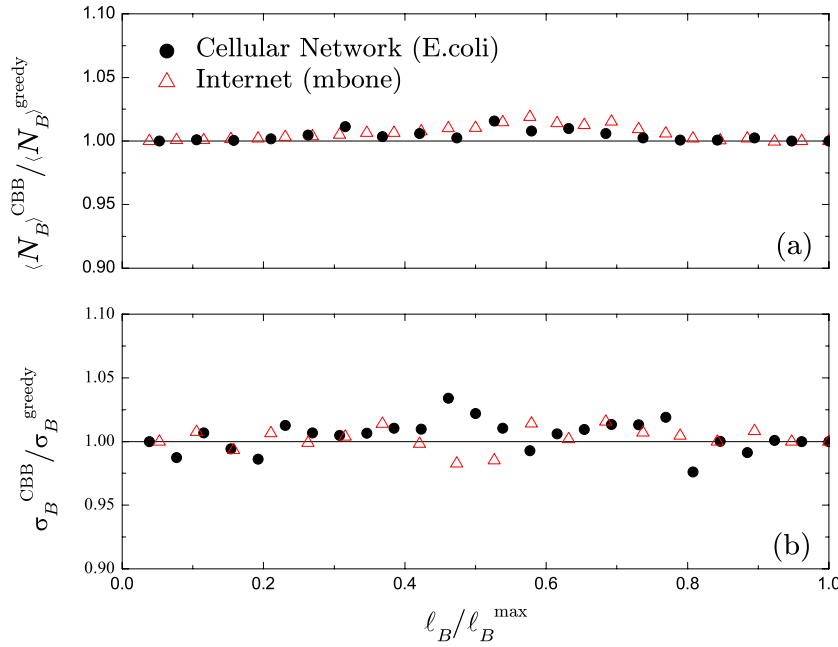
Figure 8. Comparison of (a) the mean number of boxes, $\langle N_B \rangle$, and (b) the normalized variance, σ_B, between the greedy algorithm and CBB.

where a box can be defined as nodes within a radius r_B from a central node. Using this box definition and random central nodes, this burning algorithm yields the optimal solution for non-scale-free homogeneous networks, since the choice of the central node is not important. However, in inhomogeneous networks with wide-tailed degree distribution, such as the scale-free networks, this algorithm fails to achieve an optimal solution because of the hubs existence. For example, figure 9 demonstrates that burning with the radius from non-hubs is much worse than burning from hubs. In scale-free networks, when
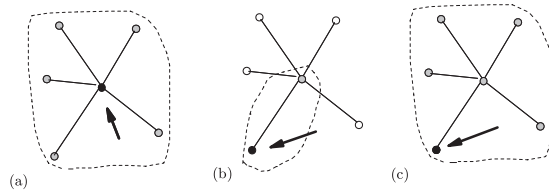
Figure 9. Burning with the radius r_B from (a) a hub node or (b) a non-hub node results in very different network coverage. In (a) we need just one box of $r_B = 1$ while in (b) five boxes are needed to cover the same network. This is an intrinsic problem when burning with the radius. (c) Burning with the maximum distance ℓ_B (in this case $\ell_B = 2r_B + 1 = 3$) we avoid this situation, since independently of the starting point we would still obtain $N_B = 1$.

selecting a random node there is a high probability that this node will not be a hub, but a low degree node instead, which leads the network tiling far from the optimal case. Additionally, a box burning originating from a non-hub node is not compact, in the sense that this box could contribute to a more efficient covering by incorporating more uncovered nodes without violating the maximum distance criterion. A variation of this algorithm for complex networks was presented in [19]. In general, this method cannot directly provide the optimum coverage, but it was shown that it finally yields the same fractal exponent d_B as the greedy colouring algorithm. Since the most important feature of similar studies is usually the calculation of the d_B exponent this algorithm can be very useful and, moreover, it is by far the easiest to implement.

To improve this completely random approach, we suggest an alternative strategy that attempts to locate some optimal 'central' nodes which will act as the burning origins for the boxes. In principle one could use the hubs as box centres. However, depending on the nature of the network, choosing the hubs may not lead to the optimal solution because the hubs may be directly connected to each other or share a large number of common nodes, and this choice practically prohibits any low degree node to be a box centre which in some cases may be beneficial. Burning from the hubs represents a special case of the method that we will present, and it may emerge naturally from this algorithm if this is indeed the optimal way to cover the network. This is the case when hubs are not directly connected. In the following algorithm we use the basic idea of box optimization, where we require that each box should cover the maximum possible number of nodes. For a given burning radius r_B, we define the 'excluded mass' of a node as the number of uncovered nodes within a chemical distance less than r_B. First, we calculate the excluded mass for all the uncovered nodes. Then we seek to cover the network with boxes of maximum excluded mass. The details of this algorithm, which we call a maximum-excluded-mass-burning or MEMB algorithm, are as follows (see figure 10):

(i) Initially, all the nodes are marked as uncovered and non-centres.

(ii) For all non-centre nodes (including the already covered nodes) calculate the excluded mass, and select the node p with the maximum excluded mass as the next centre.

(iii) Mark all the nodes with chemical distance less than r_B from p as covered.

(iv) Repeat steps (ii) and (iii) until all nodes are either covered or centres.
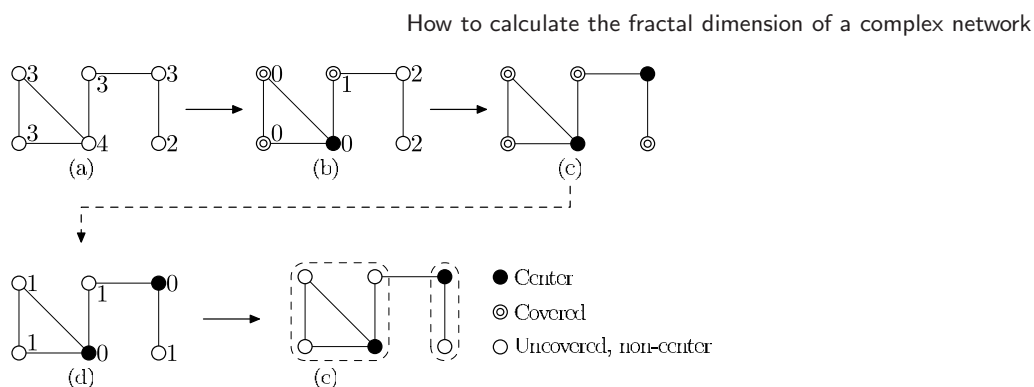
Figure 10. Illustration of the MEMB algorithm for $r_B = 1$. *Upper row: Calculation of the box centres.* (a) We calculate the excluded mass for each node. (b) The node with maximum mass becomes a centre and the excluded masses are recalculated. (c) A new centre is chosen. Now, the entire network is covered with these two centres. *Bottom row: calculation of the boxes.* (d) Each box includes initially only the centre. Starting from the centres we calculate the distance of each network node to the closest centre. (e) We assign each node to its nearest box.

Notice that the excluded mass has to be updated in each step because it is possible that it has been modified during this step. A box centre can also be an already covered node, since it may lead to a largest box mass. After the above procedure, the number of selected centres coincides with the number of boxes N_B that completely cover the network. However, the non-centre nodes have not yet been assigned to a given box. This is performed in the next step:

(i) Give a unique box id to every centre node.

(ii) For all nodes calculate the 'central distance', which is the chemical distance to its nearest centre. The central distance has to be less than r_B, and the centre identification algorithm above guarantees that there will always exist such a centre. Obviously, all centre nodes have a central distance equal to 0.

(iii) Sort the non-centre nodes in a list according to increasing central distance.

(iv) For each non-centre node i, at least one of its neighbours has a central distance less than its own. Assign to i the same id with this neighbour. If there exist several such neighbours, randomly select an id from these neighbours. Remove i from the list.

(v) Repeat step (iv) according to the sequence from the list in step (iii) for all non-centre nodes.

For both the greedy colouring and the CBB algorithm the connectivity of boxes is not guaranteed. That is, for some boxes there may not exist a path inside the box that connects two nodes belonging in this box. The reason is that some boxes may already include certain nodes that are crucial for the optimization of other boxes. The MEMB algorithm, though, always yields connected boxes and this is the most appropriate method when this condition is required.

The MEMB algorithm is nearly deterministic, especially in the calculation of the N_B value. Randomness only enters in the order of choosing two nodes at equal distance
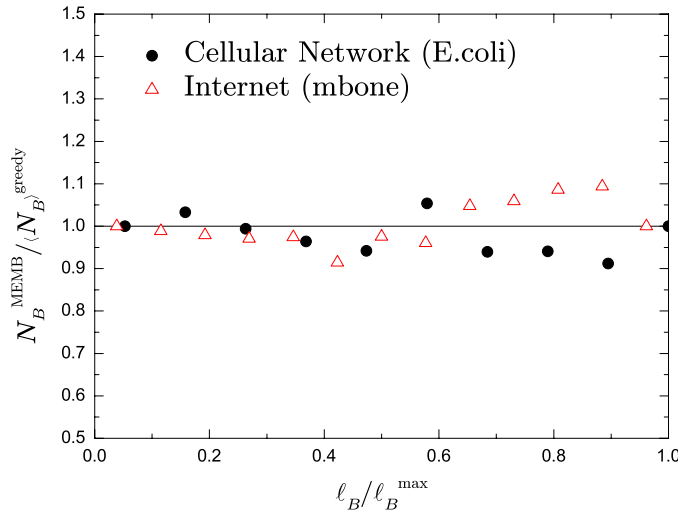
Figure 11. Comparison between the number of boxes obtained using MEMB, $N_{\mathrm{B}}^{\mathrm{MEMB}}$, and the mean number of boxes, $\langle N_{\mathrm{B}}\rangle^{\mathrm{greedy}}$, obtained from the greedy algorithm.

from two centres. In order to directly compare the results with the greedy algorithm, we convert the radius r_{B} to the box size ℓ_{B}, according to $\ell_{\mathrm{B}} = 2r_{\mathrm{B}} + 1$. Figure 11 shows that the calculated number of boxes using MEMB, $N_{\mathrm{B}}^{\mathrm{MEMB}}$, is also very similar to the mean value obtained from the greedy algorithm, $\langle N_{\mathrm{B}}\rangle^{\mathrm{greedy}}$.

The MEMB algorithm was used in figures 2 and 3 of [7] for the calculation of hub–hub correlations, because in this case we want to isolate hubs in different boxes, a behaviour similar to the model introduced in that paper (through the quantity $\mathcal{E}(\ell_{\mathrm{B}})$ defined in [7]). Also, we used this algorithm for studying the evolution of conserved proteins in the yeast protein interaction network [28].

3.3. Comparison between the different algorithms

A comparison between the greedy colouring, the CBB and MEMB algorithms with the simple completely random burning with r_{B} (figure 12) shows that the three methods, except the random burning with r_{B}, are not sensitive to the specific realization used. This is manifested in the very narrow distributions of N_{B} and in the minimum value of the distribution which is very similar in all three cases (and very close to the average value, as well). On the contrary, when we use the random burning algorithm with r_{B} the corresponding distribution is significantly wider and the mean value $\langle N_{\mathrm{B}}\rangle$ is much larger. Thus, a very large number of different realizations is required for achieving the optimal coverage in this case. Although the distributions in figure 12 correspond to a given value of ℓ_{B} (or equivalently r_{B}) the results are very similar for other ℓ_{B} values.

Despite these differences, the calculation of the fractal dimension d_{B} yields the same value for all the presented algorithms (figure 13), indicating that the scaling of the number of boxes is quite stable in all cases. Still, for the random burning it is not clear how many different realizations are needed in order for the average value to stabilize. Although from a practical point of view burning with r_{B} can still be used and give the correct dimension
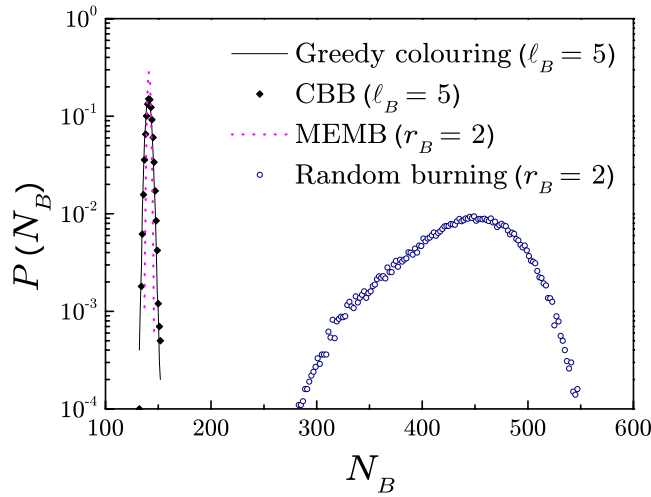
Figure 12. Comparison of the distribution of N_B for 10^4 realizations of the four network covering methods presented in this paper. Notice that three of these methods yield very similar results with narrow distributions and comparable minimum values, while the random burning algorithm fails to reach a value close to this minimum (and yields a broad distribution).
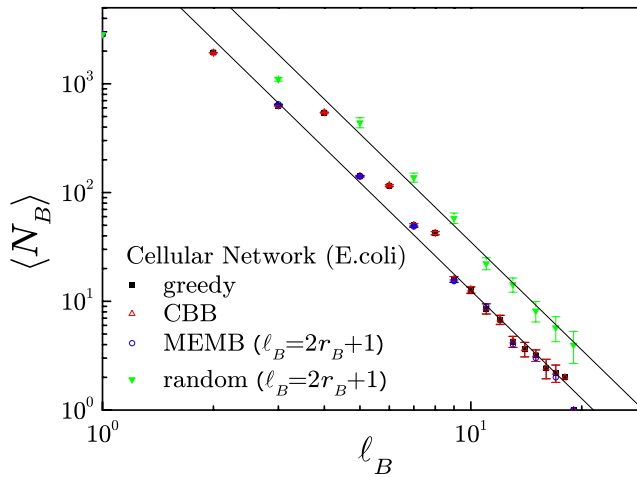

Figure 13. Comparison of the mean number of boxes $\langle N_B \rangle$ versus ℓ_B for the four presented algorithms. All methods yield the same value of the fractal dimension $d_B = 3.5$.

exponent d_B, it is not clear whether the properties of the boxes will be the same as in the optimal covering, e.g. whether applying renormalization to a network based on this covering will be similar to the renormalized network obtained from the optimal tiling.

4. Box size correction

In the usual box covering techniques applied to regular fractals, as well as in all the methods described above, the box size ℓ_B denotes the maximum possible distance within
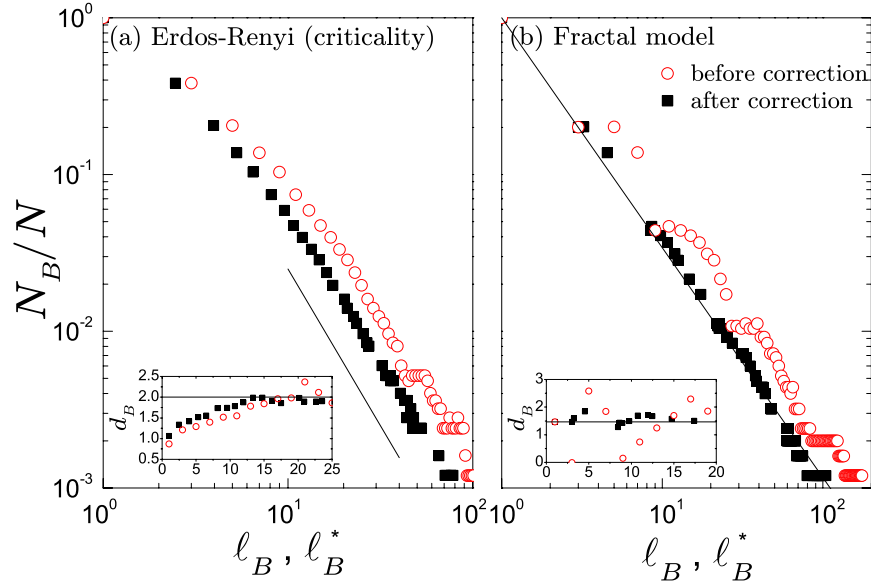
Figure 14. Comparison of the fractal dimension before and after applying the box size correction in (a) an Erdos–Renyi model at criticality and (b) a fractal network model. The straight lines correspond to the analytical predictions. Insets: improvement of the fractal dimension d_B calculation as we increase the number of boxes used for this calculation.

a box. Thus, it is always introduced as a cut-off value, rather than a direct measurement. Although in homogeneous systems, such as regular fractals, the difference may be indistinguishable, in many cases concerning inhomogeneous networks the actual size of boxes can be much smaller than this cut-off value ℓ_B. This difference is not expected to modify the asymptotic behaviour of the scaling form $N_B \sim \ell_B^{-d_B}$. However, measurement of the fractal dimension d_B in real-world networks usually requires faster convergence, due to the small-world nature of many of them. Thus, we introduce an alternative definition for the box size ℓ_B^*. This parameter corresponds now to the actual box size (after we perform the network coverage in the usual way), and is defined as the maximum distance inside the particular box plus one, which is of course always smaller than or equal to ℓ_B. The average box size ℓ_B^* over all boxes is used as a replacement of the previous cut-off size ℓ_B, and we replot the number of boxes $N_B(\ell_B^*)$ (whose maximum diameter is still ℓ_B) versus the average diameter ℓ_B^*. However, in order to obtain the correct box size and be consistent with the ℓ_B^* definition, the boxes have to be connected. Thus, we measure $N_B(\ell_B^*)$ via the MEMB algorithm, as described above.

We test the improvement of this modification by applying the measurement of ℓ_B^* to a couple of known examples. The fractal dimension of Erdos–Renyi networks at criticality ($\langle k \rangle = 1$) is known to be $d_B = 2$ (see e.g. [29]). In figure 14(a) we compare the numerical results before and after the size correction in such a network. The measurement of fractality after the correction seems to converge faster to the analytical prediction than the previous measurement. The improvement can be assessed by the inset plot where it is shown that the theoretically predicted value is achieved at smaller ℓ_B^*. Furthermore, the proposed correction has smoothed the tail in the plot, which may be crucial for

the accurate determination of d_{B}, especially for the small box sizes considered in real networks.

The improvement achieved is more prominent in the case of the fractal network model proposed in [7]. Due to the construction process of this model, this network is highly modular with very inhomogeneous distribution of the links in the modules. As a result, the number of boxes for a given size ℓ_{B} fluctuates significantly and, as shown in figure 14(b), it is very difficult to extract a reliable slope from the data. This discrete character has also been pointed in [20] where it is interpreted in terms of log-periodic oscillations in N_{B}. The use of ℓ_{B}^{*}, though, leads to a very robust slope which is exhibited over almost the entire range. As can be seen in the inset, d_{B} is practically always equal to its theoretical value when using the corrected value, in contrast to the uncorrected calculation where the value of d_{B} is more difficult to estimate.

5. Summary

In conclusion, we have shown that the box covering method is equivalent to vertex colouring in arbitrary networks. On the basis of this result, we proposed a greedy algorithm for box covering, which was found to be very accurate. A detailed analysis of the method was performed to estimate the uncertainty of the algorithm. We also introduced two geometric algorithms and compared them with the greedy algorithm. We find that all of them result in a similar optimal number of boxes. Finally, we showed that an alternative definition of the box size ℓ_{B} can lead to a more precise measurement of a network's fractal dimension.

Acknowledgments

References

[1] Albert R and Barabasi A L, 2002 *Rev. Mod. Phys.* **74** 47
[2] Dorogovtsev S N and Mendes J F F, 2002 *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford: Oxford University Press)
[3] Newman M E J, 2003 *SIAM Rev.* **45** 167
[4] Pastor-Satorras R and Vespignani A, 2004 *Evolution and Structure of the Internet: A Statistical Physics Approach* (Cambridge: Cambridge University Press)
[5] Boccaletti S, Latora V, Moreno Y, Chavez M and Hwang D U, 2006 *Phys. Rep.* **424** 175
[6] Song C, Havlin S and Makse H A, 2005 *Nature* **433** 392
[7] Song C, Havlin S and Makse H A, 2006 *Nat. Phys.* **2** 275
[8] Yook S H, Radicchi F and Meyer-Ortmanns H, 2005 *Phys. Rev.* E **72** 045105
[9] Palla G, Derenyi I, Farkas I and Vicsek T, 2005 *Nature* **435** 814
[10] Zhao F C, Yang H J and Wang B H, 2005 *Phys. Rev.* E **72** 046119
[11] Goh K I, Salvi G, Kahng B and Kim D, 2006 *Phys. Rev.* E **96** 018701
[12] Xu P, Yu B M, Yun M J and Zou M Q, 2006 *Int. J. Heat Mass Transfer* **49** 3746
[13] Barriere L, Comellas F and Dalfo C, 2006 *J. Phys. A: Math. Gen.* **39** 11739
[14] Carmi S, Havlin S, Kirkpatrick S, Shavitt Y and Shir E, 2006 *Preprint* cond-mat/0601240
[15] Moreira A A, Paula D R, Costa Filho R N and Andrade J S Jr, 2006 *Phys. Rev.* E **73** 065101
[16] Estrada E, 2006 *Europhys. Lett.* **73** 649
[17] Guida M and Maria F, 2007 *Chaos Solitons Fractals* **31** 527
[18] Jensen R T and Toft B (ed), 1995 *Graph Coloring Problems* (New York: Wiley–Interscience)
[19] Kim J S, Goh K I, Salvi G, Oh E, Kahng B and Kim D, 2006 *Preprint* cond-mat/0605324

[20] Zhou W X, Jiang Z Q and Sornette D, 2006 *Preprint* cond-mat/0605676
[21] Peitgen H O, Jurgens H and Saupe D, 1993 *Chaos and Fractals: New Frontiers of Science* (Berlin: Springer)
[22] Feder J, 1988 *Fractals* (Berlin: Springer)
[23] Bunde A and Havlin S (ed), 1995 *Fractals in Science* (Berlin: Springer)
[24] Garey M and Johnson D, 1979 *Computers and Intractability: A Guide to the Theory of NP-Completeness* (New York: Freeman)
[25] Christofides N, 1971 *Comput. J.* **14** 38
[26] Wilf H, 1984 *Inf. Process. Lett.* **18** 119
[27] Cormen T H, Leiserson C E, Rivest R L and Stein C, 2001 *Introduction to Algorithms* (Cambridge, MA: MIT Press)
[28] Song S and Makse H A, 2007 unpublished
[29] Braunstein L A, Buldyrev S V, Cohen R, Havlin S and Stanley H E, 2003 *Phys. Rev. Lett.* **91** 168701