2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Differential Evolution Box-covering Algorithm for Fractal Dimension on Complex Networks

Li Kuang, Zhiyong Zhao, Feng Wang*, *Member, IEEE*, Yuanxiang Li, Fei Yu, and Zhijie Li
State Key Laboratory of Software Engineering
Computer School, Wuhan University
Wuhan, P.R. China
fengwang@whu.edu.cn

***Abstract*—The fractality property are discovered on complex networks through renormalizaiton procedure, which is implemented by box-covering method. The unsolved problem of box-covering method is finding the minimum number of boxes to cover the whole network. Here, we introduce a differential evolution box-covering algorithm based on greedy graph coloring approach. We apply our algorithm on some benchmark networks with different structures, such as a E.*coli* metabolic network, which has low clustering coefficient and high modularity; a Clustered scale-free network, which has high clustering coefficient and low modularity; and some community networks (the Politics books network, the Dolphins network, and the American football games network), which have high clustering coefficient. Experimental results show that our algorithm can get better results than state of art algorithms in most cases, especially has significant improvement in clustered community networks.**

Keywords*—*fractal network, fractal dimension, differential evolution algorithm, box-covering algorithm

I. INTRODUCTION

Nowadays, more and more researchers pay attentions on the study of complex networks, especially on the statistical mechanism and topological structure[1][2][3]. In order to reveal the underlying topological features, many researchers have studied the fractality property of complex networks. In 2005, Song *et al* proposed a renormalization procedure to tile networks into boxes with a given box size[4]. The renormalization procedure is implemented by box-covering method, which is inspired by 'box counting' method in Euclid space[5]. Through the procedure, they found that many real world networks have fractality property, such as the World Wide Web, the Protein-Protein Interaction networks and the Cellular networks. The fractality property is defined as[4]:

$$N_B(\ell_B) \approx \ell_B{}^{-d_B}, \tag{1}$$

where $N_B(\ell_B)$ means the minimum number of boxes to cover the whole network with the box size ℓ_B, and d_B is defined as the fractal dimension. The physical importances of fractality have been extensive studied in last decades[6], [7], [8].

However, there is an unsolved problem in box-covering method: given a box size ℓ_B, how to cover a network with the minimum number of boxes. Finding the best solution of this problem is known as NP-hard[9]. Many algorithms have been applied to the box-covering problem, such as: greedy graph coloring algorithm[10], which transfer the box-covering problem to graph coloring problem; the maximum-excluded-mass-burning algorithm[10], which ensures the nodes are connected inside a box; the random burning method[11], in which random node is selected as the seed of a box and burning neighbouring nodes into this box step by step; edge covering method was applied based on simulated annealing [12]; and also merging algorithm [13]. These algorithms obtain similar results, and the graph coloring algorithm is widely used due to its low time complexity and simplicity on implementation. Later on, Schneider *et al* proposed an optimization algorithm [14]. It apply burning approach to create all possible boxes, then reduce the unnecessary boxes step by step. After the reduction, it split the network to subnetworks and applied sub-algorithms to each subnetwork. This algorithm can get significant improvement on many real networks. However, this algorithm is not well suitable for clustered community networks.

In this paper, we propose a differential evolution box-covering (DEBC) algorithm, which is based on the greedy graph coloring approach. The DEBC algorithm is independent of the network structure, it can get steady improvement in clustered community networks. We compare DEBC algorithm with standard greedy coloring algorithm and the Schneider's algorithm on 5 benchmarks networks. The results show that DEBC algorithm has significant improvement compared with greedy algorithm, and gets similar results with Schneider algorithm in most cases. Especially in clustered community networks, DEBC algorithm has obvious improvement compared with the Schneider algorithm.

This paper is organized as follows. In Section.II, we give the details of how to transfer box-covering problem to graph coloring problem. In Section.III, we introduce the structure of traditional differential evolution algorithm. In Section.IV, we show the application of differential evolution algorithm on box-covering problem. In Section.V, box-covering results comparison of 3 algorithms on 5 networks are presented. At last, we conclude our works in Section.VI.

II. BOX-COVERING TO GRAPH COLORING

The box-covering method is defined as to tile a network with minimum number of non-overlapping boxes. All the boxes have the same box size ℓ_B. The ℓ_B is the upper bound of shortest paths between all pair of nodes in each box. As we only consider unweighed and undirected networks, each pair of connected nodes have chemical distance equals to 1. And the Dijkstra's and Floyd-Warshall algorithm [15], [16] are employed to find the all-pair shortest paths.

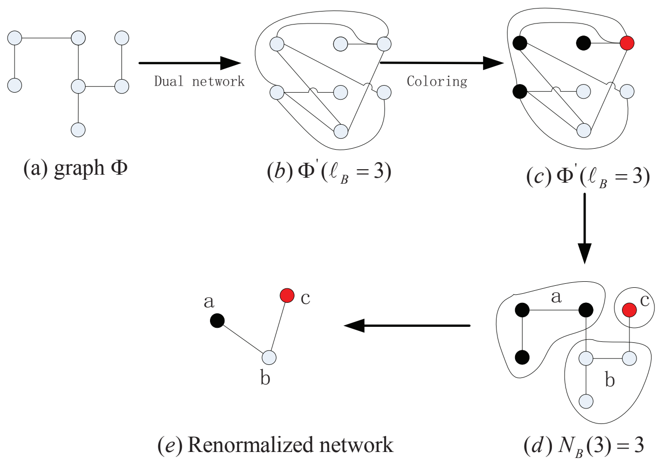


Fig. 1. Transforming box-covering problem to the graph coloring problem. We choose initial graph Φ in **(a)** to construct the dual network $\Phi^{'}$ in **(b)** for a given box size ℓ_B = 3, where two nodes will be connected if their distance not less than ℓ_B. The box-covering problem of graph Φ are transfer to graph coloring problem in $\Phi^{'}$, in which two connected nodes must have different colors as shown in **(c)**. Thus, the chromatic number of $\Phi^{'}$ is the least number of boxes needed to covering graph Φ as shown on sub-figure **(d)** and **(e)**.

The box-covering problem could be transferred into a graph coloring algorithm of its dual network $\Phi^{'}$ [10]. Graph coloring is defined as finding the chromatic number (the minimum number of colors needed to color the whole network) of the network, according to rule that two connected nodes must have different colors. The dual network $\Phi^{'}$ is constructed by: two nodes will be connected if the shortest distance between them not less than box size ℓ_B. As shown in Fig.1, given a graph Φ with 7 nodes, we can get its dual networks $\Phi^{'}$ with box size $\ell_B = 3$. Then, the greedy algorithm is applied to coloring the dual networks $\Phi^{'}$. The chromatic number is $N_B(3) = 3$. Therefore, the fractal dimension of network is calculated by iteratively finding the chromatic number of dual network $G^{'}(\ell_B)$ from ℓ_B=1 to $\ell_B = \ell_{max}$.

III. Differential Evolution Algorithm

The Differential Evolution (DE) algorithm is a new evolutionary technique introduced by Storn R and Price K [17], [18]. Due to its effectiveness and simple mathematical structure, DE has been widely used in complex function optimization, neural networks training and data mining.

The DE algorithm has three main steps in the whole generation: mutation, crossover and selection [17]. There are several kinds of DEs vary according to different mutation strategies. In this paper, we use $DE/rand/1$ strategy. In particular, for a search problem in a *D-dimensional* space, a population consists of *NP* number of parameter vectors $X_{i,G}, i = 1, 2, \cdots, NP$, where G denotes one generation, and *NP* is the number of members in a population. The fitness of the problem is represented by $F(X_{i,G})$. The new vectors $X'_{i,G}$ is generated by the processes of mutation and crossover. Thereafter, the better one between new and original vector will be chosen by the process of selection.

A. Mutation

The main difference between DE algorithm and other evolution algorithms is the process of mutation. The new vectors $V_{i,G+1}$ is generated according to [17]:

$$V_{i,G+1} = X_{r1,G} + F * (X_{r2,G} - X_{r3,G}), \quad r_1 \neq r_2 \neq r_3 \neq i, \tag{2}$$

where r_1, r_2, r_3 are randomly chosen within the interval $[1, NP]$, F is a constant control parameter controlling the amplification of the difference between vectors $(X_{r2,G} - X_{r3,G})$.

B. Crossover

In order to increase the diversity of the population, the crossover process generates the new vectors according to the crossover probability *CR*. The new vectors are generated as follows [17].

$$U_{i,G+1} = (u_{i,1,G+1}, u_{i,2,G+1}, \cdots, u_{i,D,G+1}), \tag{3}$$

where $u_{i,j,G+1}$ is generated according to:

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1}, & if\,(rand_j\,[0,1] \leq CR)\, or\,(j = j_{rand}) \\ x_{i,j,G}, & otherwise \end{cases}, \tag{4}$$

where $j = 1, 2, \cdots, D$, and $rand_j$ is a random number within the interval $[0, 1]$.

C. Selection

In the selection process, the program chooses the better vector between the original vector $X_{i,G}$ and the crossover vector $U_{i,G+1}$ by applying the greedy strategy, and puts it into the population of next generation. The select operation is defined as follows [17].

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, & if\,(f\,(U_{i,G+1}) \leq f\,(X_{i,G})) \\ X_{i,G}, & otherwise \end{cases}, \tag{5}$$

where $f\,(U_{i,G+1})$ is the fitness of the $U_{i,G+1}$, and $f\,(X_{i,G})$ is the fitness of the $X_{i,G}$.

IV. Differential Evolution Box-covering (DEBC) Algorithm

There are several kinds of algorithms which have been applied to solve the box-covering problem. As we introduced in Section.II, the box-covering problem can be transferred into a graph coloring problem of its dual network. The deterministic greedy graph coloring algorithm is widely used [10]. It is an effective method, but it usually get results far from the global optimum solution and easy to stuck in local optimum solution. By evolutionary strategy which mentioned in Section.III, the DE algorithm can avoid this drawback, and find the solution close to global optimum. In this section, we present our DEBC algorithm, which based on deterministic greedy graph coloring algorithm.

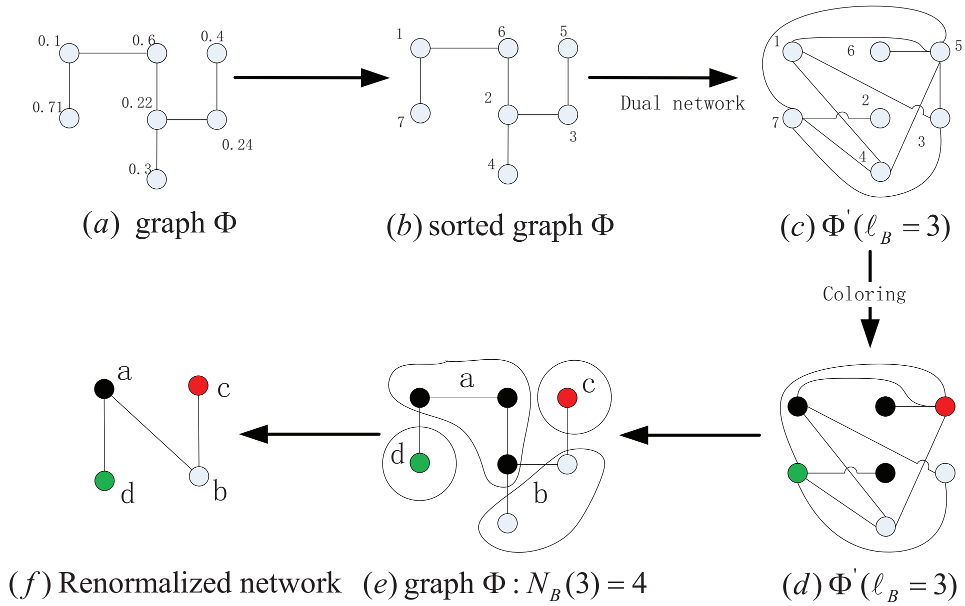


Fig. 2. Greedy graph coloring to ordered network. **(a)**, We give each node in initial graph Φ a random number between $[0, 1]$. **(b)**, We sort the nodes by the numbers they have by ascending order and assign each node an Order-ID. **(c)**. Construct the dual network $\Phi^{'}$ with given box size ℓ_B = 3. **(d)**, We apply greedy algorithm to the dual network $\Phi^{'}$ following the order of nodes. That means each node only compare with nodes with smaller node Order-ID. **(e)** and **(f)**, we get $N_B(3) = 4$ differs from the result shown in Fig.1. It shows different node orders lead to different N_B.

A. Individual Encoding on greedy algorithm

As shown in Fig.2, the order of the nodes in the graph determines the minimum number of colors used to color a graph [19]. Each individual represents a node order sample of a graph. The individual of the population in DE algorithm can be represented as $X_{i,G} = \{x_{i,1}, x_{i,2}, \cdots x_{i,D}\}, i = 1, 2, \cdots, NP$, where G denotes one generation, D stands for the number of the nodes in the graph, and $x_{i,D}$ is a random number within the interval $[0, 1]$, which represents the order of the ***D-th*** node. The pseudocode of the greedy coloring algorithm is shown as Algorithm.1 [10].

B. Fitness definition and Main DEBC algorithm

The fitness of individual in DEBC is defined as equation(6):

$$F_{debc}(X_{i,G}) = N_B(X_{i,G}), \tag{6}$$

where $N_B(X_{i,G})$ is the number of the colors needed to color the individual $X_{i,G}$ by greedy coloring algorithm. In each generation, we firstly use the crossover and mutation operators to change the sequence of the vertices, then measure the fitness of all individuals by greedy coloring algorithm, at last we select the best sequence up to now by greedy selection strategy. The pseudocode of DEBC algorithm is shown as Algortihm.2.

C. Complexity of DEBC algorithm

Among the three algorithm we applied, the fastest is the greedy coloring algorithm, which has time complexity of $O(N^2)$ on a network with N nodes. And the required time complexity of Schneider algorithm depends on the network structure [14]. For tree networks it could be performed in $O(N^3)$, while for regular networks it requires $O(2^N)$.

Algorithm 1 Greedy Coloring Algorithm

Require: $X_{i,G}$ is an vector that encodes a node order of a network. N is the number of nodes in the network. n and j are variables represent Order-ID of a node.
Ensure: minimum $N_B(X_{i,G})$

```
Sort the value of X_{i,G} by ascending order, and assign a unique Order-ID from 1 to N to all nodes according to the order of the nodes.
Assign a color c = 0 to node with Order-ID n = 1, create a color set Set_c, add c = 0 to Set_c.
for n = 2 to N do
    Create a local color set Set_n
    for j = 1 to n − 1 do
        If node j is connected with node n in dual network Φ', add the colors of node j to Set_n.
    end for
    If set (Set_c − Set_n) contains colors, random select one color from (Set_c − Set_n) to assign to node n; else if (Set_c − Set_n) is empty, create a new color c, assign c to node n and add c to Set_c
end for
Count the colors in Set_c, the number of colors is same as box number N_B(X_{i,G}).
```

As for DEBC algorithm, the most time consuming step is the fitness calculation. The time of fitness calculation is divided to two parts: (1) sorting the individual $X_{i,G}$, which takes $O(N \log N)$, (2) applying the greedy coloring algorithm, which requires $O(N^2)$. Thus, the time complexity of fitness calculation is $O(N^2)$. Therefore, the total computational time of DEBC algorithm is

$$G_{max} * NP * O(N^2), \tag{7}$$

Algorithm 2 DEBC algorithm

Require: G denotes the generation number, NP denotes the number of population, F denotes the control parameter in the mutation step, CR denotes the control parameter in the crossover step, and G_{max} is the maximum generation.

Ensure: optimal F_{debc}

```
Set G = 0, NP, F, CR, ℓ_B, and random initialize X_{i,G}.
Find the dual network Φ' for given ℓ_B by calculating the all-pair shortest paths.
Calculate the fitness of each individual F_debc(X_{i,0}) by greedy coloring algorithm
repeat
    The Mutation phase:
    for i = 1 to NP do
        V_{i,G+1} = X_{r1,G} + F * (X_{r2,G} − X_{r3,G}), r1 ≠ r2 ≠ r3 ≠ i.
    end for
    The Crossover phase:
    for i = 1 to NP do
        U_{i,G+1} = (u_{i,1,G+1}, u_{i,2,G+1}, ··· , u_{i,D,G+1}).
    end for
    Calculate the fitness of each individual F_debc(X_{i,G}) by greedy coloring algorithm
    The Selection phase:
    X_{i,G+1} = { U_{i,G+1},  if(F_debc(U_{i,G+1}) ≤ F_debc(X_{i,G}))
                { X_{i,G},    otherwise
until (G > G_max)
```

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, & if(F_{debc}\left(U_{i,G+1}\right) \leq F_{debc}\left(X_{i,G}\right)) \\ X_{i,G}, & otherwise \end{cases}.$$

where G_{max} is the maximum number of generation and NP is the number of generation. In evolutionary algorithms, each individual can be calculated independently. Thus, the parallel processes can be applied to reduce the computational time to $G_{max} * O(N^2)$.

V. Experiments and Results

In order to show the effeteness of our DEBC algorithm, in this section, we show the $N_B(\ell_B)$ results of DEBC algorithm on 5 benchmark networks and compared the results with the greedy graph coloring algorithm(greedy) [10] and Schneider's box-covering algorithm(Sch) [14]. Although we only apply undirected, unweighted networks, this DEBC algorithm can be easily extended to the directed and weighted networks [20].

A. Benchmark Networks

Here, we apply five benchmark networks with different structures to show the structure independence of our DEBC algorithm. The properties of benchmark networks are shown in Table.I. Firstly, we apply the widely used biological network E.*coli*, which has low clustering coefficient and high modularity.

(1) The Escherichia coli network (E.*coli*): Protein-protein Interaction network with 2859 proteins and 6890 interactions between them [21].

In order to show the improvements of DEBC algorithm on highly clustered coefficient networks, we apply the the Clustered scale-free network, which has high clustering coefficient and low modularity.

(2) The Clustered scale-free network (CSF): The modified Barabási-Albert model with high clustering coefficient [22]. Here, the applied CSF network has 2003 nodes and 8000 edges.

Since the CSF network has low modularity. We also applied three clustered community networks (the polbooks, dolphin, and football networks) as follows. These community networks have both high modularity and clustering coefficient.

(3) The Politics books network (polbooks): A network of books about US politics with 105 books and 441 edges. Edges between books represent frequent copurchasing of books by the same buyers [23].

(4) The Dolphins network (dolphin): An social network of 62 dolphins in a community. Edges represent frequent associations between dolphins [24].

(5) The American football games network (football): Football games between 115 colleges during season 2000 [25]. Each node represents a college football team and each edge represents a match between two attached teams.

Table.I also illustrates the average improvement of Schneider algorithm and DEBC algorithm compared with the greedy box-covering algorithm. The $Avg(\Delta N_{Sch}/N_{greedy})$ and $Avg(\Delta N_{DEBC}/N_{greedy})$ are defined as the average of relative improvements to the greedy box-covering algorithm at all length scales of ℓ_B. It shows DEBC algorithm gets better average results than Schneider algorithm in all benchmark networks, especially in clustered community networks.

These three algorithms have different box definitions: the Schneider algorithm uses the radius (r_B) to draw a box, while the greedy algorithm and our DEBC algorithm use the maximum shortest path (ℓ_B). The radius holds more strict definition than box size ℓ_B, which r_B can be transferred to ℓ_B as $\ell_B = 2r_B + 1$. Due to the deference, the Schneider algorithm is easy to fall into local optimum solution for networks with both high clustering coefficients and modularity, such as the political books network, the dolphin network, and the football network. Therefore, the steady improvements of DEBC algorithm in different kinds of networks shows that it is independent of network structures.

B. Result Analysis

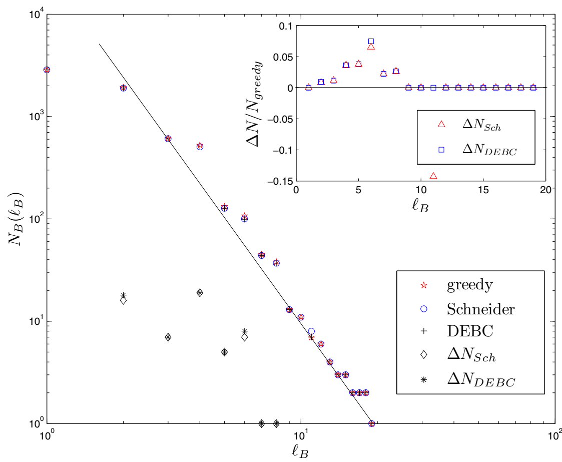
Here, we illustrate the results of minimum number of boxes $N_B(\ell_B)$ to cover a network in different length scales of ℓ_B as equation (1). We compare $N_B(\ell_B)$ vs ℓ_B results of DEBC algorithm with greedy and Schneider algorithms. The improvements of DEBC algorithm compared with greedy algorithm (ΔN_{DEBC}) and Schneider algorithm compared with greedy algorithm (ΔN_{Sch}) are concerned.

1) Results on the E.coli network and CSF network: As shown in Fig.3(a), for the E.*coli* network, both Schneider algorithm and DEBC algorithm have significant improvement than greedy algorithm for $\ell_B < 9$, except at $\ell_B = 11$ the Schneider algorithm is even worse than greedy algorithm. DEBC algorithm has slight improvement compared with Schneider algorithm for $\ell_B = 2, 6, 11$.

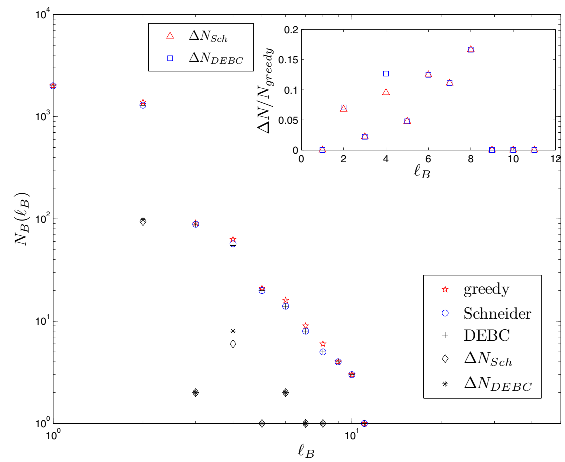
The second network is Clustered Scale-free network. As shown in Fig.3(b), both Schneider algorithm and DEBC algorithm have significant improvement up to 17% than greedy

TABLE I. PROPERTIES OF BENCHMARK NETWORKS AND AVERAGE IMPROVEMENTS OF DEBC AND SCH ALGORITHMS COMPARED WITH GREEDY ALGORITHM

Networks	N	Clustering Coefficient	Modularity	$Avg(\Delta N_{Sch}/N_{greedy})$	$Avg(\Delta N_{DEBC}/N_{greedy})$
E.coli	2859	0	0.604	0.34%	1.15%
CSF	2003	0.839	0.117	5.78%	6.09%
polbooks	105	0.488	0.527	-3.57%	7.48%
dolphins	62	0.522	0.303	-5.96%	11.61%
football	115	0.403	0.601	-60.71%	16.9%



(a) The E.*coli* network



(b) The Clustered scale-free network

Fig. 3. **(a)**,The E.*coli* Network and **(b)**, the Clustered Scale-free Network: Comparison of the minimal number of boxes $N_B(\ell_B)$ for a given box size ℓ_B using the greedy coloring algorithm, Schneider algorithm and DEBC algorithm. The two parameters $\Delta N_{Sch} = N_{greedy} - N_{Sch}$ and $\Delta N_{DEBC} = N_{greedy} - N_{DEBC}$ show the improvement of Schneider algorithm and DEBC algorithm compared with greedy algorithm, respectively. The straight line in (a) shows the power law fit of fractal dimension. The insets show the relative improvements $\Delta N/N_{greedy}$, which gives a more clear view of the improvements.

algorithm for $\ell_B < 9$. The DEBC algorithm has slight improvement compared with Schneider algorithm for $\ell_B = 2$, and has 3% improvement for $\ell_B = 4$.

2) Results on clustered community networks: Due to the different box definitions, we find the Schneider algorithm can not get optimal results in clustered community networks. In this kind of situation, DEBC algorithm shows significant improvement compared with both greedy algorithm and the Schneider algorithm. Here, we apply the three algorithms on 3 community networks: the political books network, the dolphin network, and the football network.

As shown in Fig.4(a) inset, for political books network, the Schneider algorithm is worse than greedy algorithm for $\ell_B = 2, 4$ as illustrated by $\Delta N_{Sch} < 0$. In contrast, DEBC algorithm shows steady improvement up to 33% compared with greedy algorithm for $\ell_B < 6$.

For the dolphin network, as shown in Fig.4(b), the Schneider algorithm is worse than greedy algorithm for $\ell_B = 4, 9$ as illustrated by $\Delta N_{Sch} < 0$. Instead, DEBC algorithm shows steady improvement up to 33.3% compared with greedy algorithm for $\ell_B < 8$.

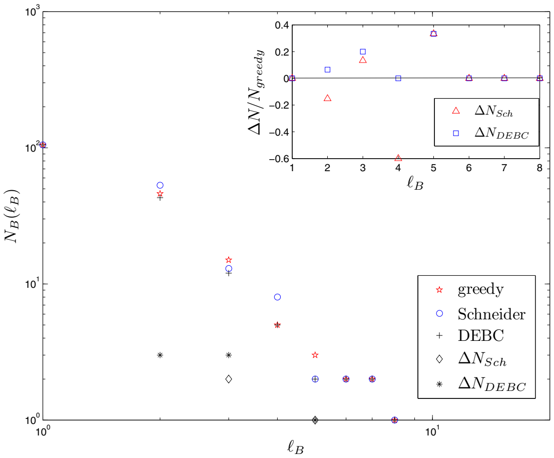
The last network is the football network. As shown in Fig.4(c), the Schneider algorithm is worse than greedy algorithm from $\ell_B = 2$ to $\ell_B = 6$ as illustrated by $\Delta N_{Sch} < 0$. However, DEBC algorithm shows steady improvement up to 45% compared with greedy algorithm for $\ell_B < 5$.

C. Discussion

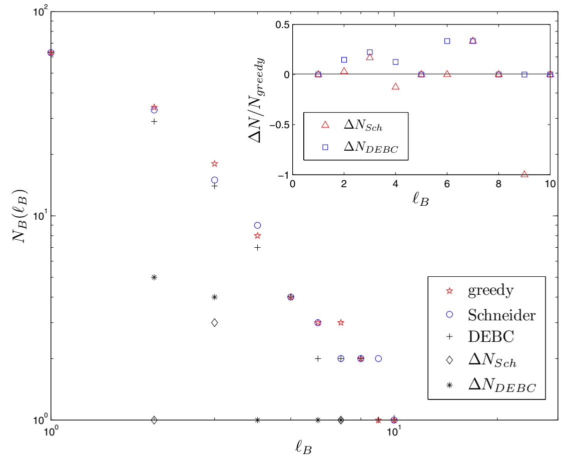
It is interesting that the improvement made by the Schneider algorithm and DEBC algorithm does not affect the fractality of networks. As the fractal dimension d_B varies a little within the tolerance. For the E.*coli* network, the best-fit of fractal dimension changes from $d_B = 3.47 \pm 0.11$ (for greedy algorithm) to $d_B = 3.45 \pm 0.10$ (for Schneider algorithm) and $d_B = 3.44 \pm 0.09$ (for DEBC algorithm).

Our experiments show that DEBC algorithm converges within a small number of generations. As shown in Fig.4(d), we pick one length scale ℓ_B for four benchmark networks, the results show that DEBC algorithm converges within the 70 generations. Due to the size of search space, the convergent rate of DEBC algorithm is inverse proportion to ℓ_B, which means the DEBC algorithm coverage faster with bigger ℓ_B. Nevertheless, all our results in this paper are reached within 5000 generation with population $NP = 40$. In our experiments, we define the mutation control parameter $F = 0.9$ and crossover probability $CR = 0.85$. And, we found that making some adjustments of these two parameters have no obvious influence on the final results.

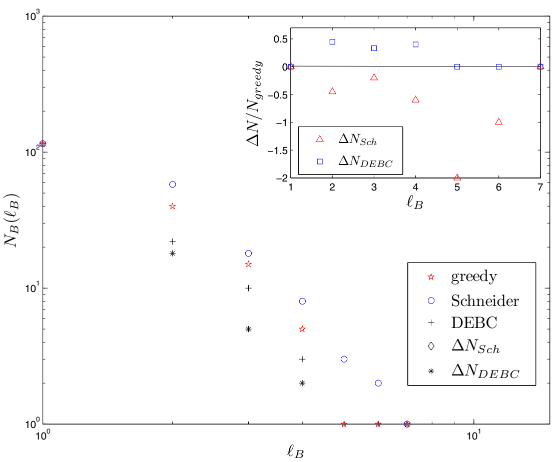
Moreover, further research is needed to minimize the search space of DEBC algorithm by reducing the unnecessary search paths. From the evolution process of DEBC algorithm, we can learn which kind of boxes are most likely to included in the global best solution. The patterns of the most likely boxes could help us redesign our algorithm, so that it can quickly adjust to the environments.



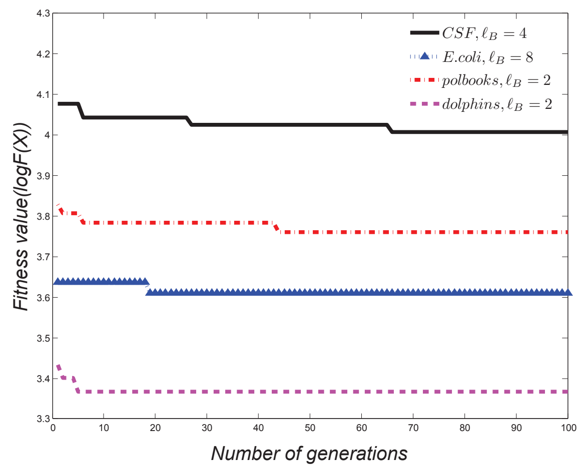
(a) The Politics books network



(b) The Dolphin social network



(c) The American football games network



(d) The Convergent Rate

Fig. 4. **(a)**,**(b)** and **(c)**, Result comparisons of the minimal number of boxes $N_B(\ell_B)$. Applying the greedy coloring algorithm, Schneider algorithm and DEBC algorithm to the Politics books network, the Dolphin social network and the Clustered Scale-free network, respectively. **(d)**,The convergent rates of our DEBC algorithm in one length scale ℓ_B of four benchmark networks. The X axis is the number of generations, and Y axis is the logarithmic of fitness.

VI. Conclusions

In this paper, we proposed a differential evolution box-covering algorithm (DEBC) to search for the minimum number of boxes to cover a network. We compared DEBC algorithm with the classical greedy graph coloring algorithm and a state of art Schneider box-covering algorithm. Our DEBC algorithm has a significant and steady improvement compared with the greedy graph coloring algorithm. It also get similar results with Schneider algorithm in most cases. Especially in clustered community networks, DEBC algorithm has obvious improvement compared with the Schneider algorithm. Moreover, due to the robustness of fractal networks, this work has practical significance on communication and transport networks construction, which have optimization objectives on robustness of the networks.

Acknowledgment

This work is supported by National Nature Science Foundation of China [grant number 61103125]; the Doctoral Fund of Ministry of Education of China [grant number 20100141120046]; the Natural Science Foundation of Hubei Province of China [grant number 2010CDB08504]; the 111 Programme of Introducing Talents of Discipline to Universities [grant number B07037]; Wuhan University Academic Development Plan for Scholars after 1970s ("Research on Internet User Behavior"); State Key Laboratory of Software Engineering (China), [grant number SKLSE2012-09-15], and the State Key Laboratory of Networking and Switching Technology (China), [grant number SKLNST-2010-1-04].

References

[1] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[2] R. A. Albert-László Barabási, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[3] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[4] C. Song, S. Havlin, and H. A. Makse, "Self-similarity of complex networks," *Nature*, vol. 433, no. 7024, pp. 392–395, Jan. 2005.

[5] B. B. Mandelbrot, "How long is the coast of britain," *Science*, vol. 156, no. 3775, pp. 636–638, 1967.

[6] H. D. Rozenfeld, C. Song, and H. A. Makse, "Small-world to fractal transition in complex networks: a renormalization group approach," *Physical review letters*, vol. 104, no. 2, p. 025701, 2010.

[7] L. K. Gallos, C. Song, S. Havlin, and H. A. Makse, "Scaling theory of transport in complex biological networks," *Proceedings of the National Academy of Sciences*, vol. 104, no. 19, pp. 7746–7751, 2007.

[8] L. K. Gallos, H. A. Makse, and M. Sigman, "A small world of weak ties provides optimal global integration of self-similar modules in functional brain networks," *Proceedings of the National Academy of Sciences*, vol. 109, no. 8, pp. 2825–2830, 2012.

[9] M. R. Gary and D. S. Johnson, "Computers and intractability: A guide to the theory of np-completeness," 1979.

[10] C. Song, L. K. Gallos, S. Havlin, and H. A. Makse, "How to calculate the fractal dimension of a complex network: the box covering algorithm," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2007, no. 03, pp. P03 006–, 2007.

[11] J. Kim, K. Goh, B. Kahng, and D. Kim, "Fractality and self-similarity in scale-free networks," *New Journal of Physics*, vol. 9, no. 6, p. 177, 2007.

[12] W.-X. Zhou, Z.-Q. Jiang, and D. Sornette, "Exploring self-similarity of complex cellular networks: The edge-covering method with simulated annealing and log-periodic sampling," *Physica A: Statistical Mechanics and its Applications*, vol. 375, no. 2, pp. 741–752, 2007.

[13] M. Locci, G. Concas, R. Tonelli, and I. Turnu, "Three algorithms for analyzing fractal software networks," *WSEAS Transactions on Information Science and Applications*, vol. 7, no. 3, pp. 371–380, 2010.

[14] C. M. Schneider, T. A. Kesselring, J. S. Andrade Jr, and H. J. Herrmann, "Box-covering algorithm for fractal dimension of complex networks," *Physical Review E*, vol. 86, no. 1, p. 016707, 2012.

[15] B. Cherkassky, A. V. Goldberg, and T. Radzik, "Shortest paths algorithms: Theory and experimental evaluation," *Mathematical Programming*, vol. 73, pp. 129–174, 1993.

[16] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[17] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[18] K. V. Price, R. M. Storn, and J. A. Lampinen, "Differential evolution a practical approach to global optimization," 2005.

[19] A. T. Hoang, G. T. T. Trinh, and V. T. Le, "A novel particle swarm optimization-based algorithm for the graph coloring problem," in *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*. IEEE, 2009, pp. 1–3.

[20] D.-J. Wei, Q. Liu, H.-X. Zhang, Y. Hu, Y. Deng, and S. Mahadevan, "Box-covering algorithm for fractal dimension of weighted networks," *Scientific reports*, vol. 3, 2013.

[21] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[22] K. Klemm and V. M. Eguiluz, "Highly clustered scale-free networks," *Physical Review E*, vol. 65, no. 3, p. 036123, 2002.

[23] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[24] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[25] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.