7-23-2008

Community detection in complex networks by dynamical simplex evolution

Vladimir Gudkov
University of South Carolina - Columbia, gudkov@sc.edu

V. Montealegre
University of South Carolina - Columbia

S. Nussinov

Z. Nussinov

Publication Info

Published in *Physical Review E*, Volume 78, Issue 1, 2008, pages 016113-1-016113-7.
Gudkov, V., Montealegre, V., Nussinov, S., and Nussinov, Z. (2008). Community detection in complex networks by dynamical simplex evolution. *Physical Review E*, 78(1), 016113-1 - 016113-7. doi: 10.1103/PhysRevE.78.016113

Community detection in complex networks by dynamical simplex evolution

V. Gudkov,[1] V. Montealegre,[1] S. Nussinov,[1,2] and Z. Nussinov[3]
[1]*Department of Physics and Astronomy, University of South Carolina, Columbia, South Carolina 29208, USA*
[2]*Tel-Aviv University, School of Physics and Astronomy, Tel-Aviv, Israel*
[3]*Department of Physics, Washington University, St. Louis, Missouri 63160, USA*
(Received 2 October 2007; revised manuscript received 15 April 2008; published 23 July 2008)

We benchmark the dynamical simplex evolution (DSE) method with several of the currently available algorithms to detect communities in complex networks by comparing correctly identified nodes for different levels of "fuzziness" of random networks composed of well-defined communities. The potential benefits of the DSE method to detect hierarchical substructures in complex networks are discussed.

DOI: 10.1103/PhysRevE.78.016113 PACS number(s): 89.75.Da, 89.75.Fb, 89.75.Hc

I. INTRODUCTION

Community detection in complex networks is attracting much interest in many areas. Partitioning a network into groups of nodes that are more tightly linked (densely connected sets of nodes) is crucial for understanding the structure, functionality, and evolution of the whole network and its building constituents. It is useful for many practical purposes including the study of real world network vulnerabilities. However, real world networks are usually very large, and community detection in complex networks is computationally very demanding [1], especially if a high level of accuracy is required [2]. Many approaches to efficient solutions of this problem have been proposed. These include spectral analysis and hierarchical clustering methods. More recently much attention has been drawn to optimization of a quantity known as modularity [2–6].

Modularity is essentially a measure of the number of links inside the detected modules of a network compared with the expected number of links that a random network with the same size and distribution of degrees would have. It has been used in several community detection methods to test the goodness of their solutions and pick up the best one. For instance, the method proposed by Girvan and Newman (GN) [5,6] uses the concept of "betweenness" to achieve a division algorithm that progressively removes links with the largest "betweenness" until the network breaks up into components. This results in several partitions, and the partition with the optimal modularity value is chosen. However, the properties of modularity have not been fully studied, and the resolution of the clustering method based upon its optimization is intrinsically limited in a manner depending on the number of links in the network [7]. The existence of a resolution limit for community detection implies that it is *a priori* impossible to tell whether a module contains substructure (that is if smaller clusters can be refined inside it). This is particularly important if the network has a self-similar character (e.g., a scale-free network), in which case a single partition does not describe the structure completely; and a tree-like partition that digs into different levels of structure is more appropriate.

II. THE ALGORITHM

In this paper we use the dynamical simplex evolution (DSE) method [8] to perform community identification in computer simulated networks and compare it with the methods mentioned above in Refs. [2–6]. The DSE algorithm has a broad range of potential applications such as testing graph isomorphism [9,10], searching the largest cliques [11], and hard graph theoretical problems in general [12]. The DSE method is particularly useful for community detection and for finding hierarchical structures in networks [13]. Some of its features are a unique solution, the preservation of all links at all stages of the calculation, and the possibility of analyzing the network's community structure at different scales without additional computation in each step of the algorithm. Let us describe the basic features of the DSE method.

A. Dynamical simplex evolution

A network consisting of n nodes connected by links can be represented by means of an $n\times n$ connectivity matrix C. If i and j are the labels of two nodes which are connected in the network, $C_{ij}=1$, and $C_{ij}=0$ if there is no link between them. In general, one can assign different values to the elements C_{ij} so as to describe the nature and intensity of connections. A single network can be represented by many matrices which are related by a similarity transformation $C'=U^{-1}CU$ with U $(U^{-1}=U^{\dagger})$ an $n\times n$ matrix affecting the permutation of rows and columns. The problem of deciding whether two connectivity matrices correspond to the same network is very difficult. We cannot try all the $n!$ permutations of the labels of the n nodes in the network in search for the particular connectivity matrix that most clearly exhibits the clusters. (We use the word cluster or community indistinctly to refer to a group of nodes which belong to a given network and which are relatively strongly interconnected yet weakly connected to nodes outside the group.)

Let us assume that a large subset of nodes in the network indeed divides into k fairly well-defined clusters $C_1,\ldots,C_k$ with $n_1,\ldots,n_k$ nodes, respectively. If the labeling is such that all nodes belonging to the same cluster are contiguous, the representative connectivity matrix is almost block diagonal: the $n_1\times n_1,\ldots,n_k\times n_k$ submatrices along the diagonal are also the connectivity matrices for the first, second, up to the kth cluster. By assumption these matrices have a larger proportion of nonzero elements than the regions of the matrix outside the diagonal blocks. This nice feature completely disappears after a massive relabeling of the nodes is performed

(i.e., massive joint reshufflings of columns and rows of the connectivity matrix). The whole matrix will then have a roughly homogeneous distribution of ones and zeroes. Our goal is essentially to reconstruct the original convenient almost block diagonal form which exhibits the clusters without resorting to combinatorial alternatives. To this end we introduce the DSE algorithm.

The n nodes are represented by n point masses in an $n-1$ dimensional space at locations $\vec{r}_i(t)$ where $i=1,\ldots,n$. To start at $t=0$ the $\vec{r}_i(t)$ in a completely symmetric and unbiased manner we place them at the n vertices of a symmetric simplex inscribed inside the unit sphere in $n-1$ dimensions so that $\vec{r}_i^2=1$ holds for all vertices. Symmetry requires all $\vec{r}_i\cdot\vec{r}_j$ to be equal for any $i\neq j$. Therefore, using $\Sigma_{i=1}^n\vec{r}_i=0$ and $\vec{r}_i^2=1$ we have

$$\left(\sum_{i=1}^{n}\vec{r}_i\right)^2=\sum_{i=1}^{n}\vec{r}_i^2+2\sum_{i>j}^{n}\vec{r}_i\cdot\vec{r}_j=n+2\frac{n(n-1)}{2}\vec{r}_i\cdot\vec{r}_j=0 \tag{1}$$

and

$$\vec{r}_i\cdot\vec{r}_j=-\frac{1}{n-1}\quad\text{for all}\quad i\neq j,\quad i,j=1,\ldots,n. \tag{2}$$

The distance between any pair of vertices of the starting simplex of the dynamical simulation is therefore

$$|\vec{r}_i-\vec{r}_j|=\sqrt{\frac{2n}{n-1}}. \tag{3}$$

The specific coordinates of the n vertices can be recursively fixed as follows. Let the vertices of the $n-1$ simplex (in $n-2$ dimensions) be $\vec{\vartheta}_1,\ldots,\vec{\vartheta}_{n-1}$ with $\vec{\vartheta}_j$ vectors with known components

$$\vec{\vartheta}_j=\sum_{k=1}^{n-2}\vartheta_{jk}\hat{e}_k, \tag{4}$$

where $\hat{e}_k$ is the unit vector along the kth axis. Then, choose the position of the nth vertex of the n-simplex (in $n-1$ dimensions) to be $\vec{r}_n=\hat{e}_{n-1}$. This requires adjusting the positions of the other $n-1$ vertices in the $n-1$ dimensional space. We write those as linear combinations of the "old" vectors $\vec{\vartheta}_i$ (in $n-2$ dimensions) and the unit vector in the new spatial direction

$$\vec{r}_i=\lambda_n\vec{\vartheta}_i-\frac{1}{n-1}\hat{e}_{n-1},\quad i=1,\ldots,n-1. \tag{5}$$

The coefficient of $-1/(n-1)$ in the second term appears because $\vec{\vartheta}_i\cdot\hat{e}_{n-1}=0$ and $\vec{r}_i\cdot\vec{r}_n=-1/(n-1)$ [from Eq. (1)]. Normalization requires $\vec{r}_i^2=1$, and since $\vec{\vartheta}_i^2=1$, we find that $\lambda_n=\sqrt{1-1/(n-1)^2}$. Thus, starting with a two simplex with $\vec{\vartheta}_1=\hat{e}_1$, and $\vec{\vartheta}_2=-\hat{e}_1$, we inductively generate the n simplex.

Next we postulate attractive "forces" $\vec{F}_{ij}$ in the $n-1$ dimensional space between point masses corresponding to pairs of nodes which are connected in the network. This groups strongly interconnected nodes, moving them to spatial proximity so as to help us identify the clusters. As the forces displace the point masses representing the nodes from their original symmetric positions the mutual distances $|\vec{r}_i-\vec{r}_j|$ keep changing and the simplex evolves.

When $C_{ij}\neq 0$ the force between the point masses $\vec{r}_i(t)$ and $\vec{r}_j(t)$ is postulated to be

$$\vec{F}_{ij}=S_{ij}f(|\vec{r}_i-\vec{r}_j|)\frac{\vec{r}_i-\vec{r}_j}{|\vec{r}_i-\vec{r}_j|}. \tag{6}$$

It acts in the direction of $\vec{r}_i-\vec{r}_j$. To retain the initial symmetry and avoid biasing we take the same force law $f(r)$ for all connected pairs.

The only way information about the specific network of interest is communicated to the dynamical n body system is via the overall strengths of the forces S_{ij}. In particular S_{ij} vanishes if $C_{ij}=0$.

The point masses can move according to Newtonian dynamics,

$$m_i\frac{d^2\vec{r}_i}{dt^2}=\vec{F}_i=\sum_j\vec{F}_{ij}. \tag{7}$$

In order to avoid "overshoots" and oscillations we add damping via viscous frictional forces, simulating the motion of the vertices as the motion of point masses in a liquid with a high viscosity,

$$m_i\frac{d^2\vec{r}_i}{dt^2}+\mu_i\frac{d\vec{r}_i}{dt}=\vec{F}_i. \tag{8}$$

If we adopt the extreme $\mu_i\gg m_i$ inertial effects become negligible and we have first order "Aristotelian dynamics." This simplifies the calculations of the subsequent positions.

Discretizing

$$\mu_i\frac{d\vec{r}_i}{dt}=\vec{F}_i \tag{9}$$

by using time increments δ we have

$$\vec{r}_i(t+\delta)=\vec{r}_i(t)+\frac{\delta}{\mu_i}\vec{F}_i(\vec{r}_i(t)),\quad i=1,\ldots,n. \tag{10}$$

To preserve the symmetry we take all masses (and separately all viscosities) to be equal $\mu_i=\mu$, $m_i=m$.

The attractive central forces can be derived from a pairwise potential, i.e.,

$$f(r)=-\frac{dU(r)}{dr}. \tag{11}$$

With overall potential energy

$$U(\vec{r}_1,\cdots,\vec{r}_n)=\sum_{i>j}S_{ij}U(|\vec{r}_i-\vec{r}_j|). \tag{12}$$

At the equilibrium positions of the dynamical system

$$\frac{d\vec{r}_i}{dt}=\vec{F}_i=\vec{0}, \tag{13}$$

and the sets of vectors $\{\vec{r}_i\}$ for which Eqs. (13) are satisfied then are stationary points of $U(\vec{r}_1,\ldots,\vec{r}_n)$. There is consider-

able freedom in choosing the force law $f(r)$. We find the simple choice of forces with constant magnitude independent of the mutual distances between vertices particularly useful. This choice tends to yield only one minimum of the multidimensional potential, and, as a consequence, a single (unique) solution.

If there are only attractive forces, the vertices rapidly collapse to a single point representing the whole network even when the vertices are constrained to stay on the surface of the unit hypersphere on which they are originally placed. The introduction of repulsive forces between point masses representing disconnected nodes reduces the rate of this collapse allowing the vertices the opportunity to cluster slowly via the attractive forces.

The maximal number of possible connections in a network is $M=n(n-1)/2$, where n is the number of nodes in the network. In practical complex networks (real world and/or simulated) the number of links is only a tiny fraction of M. For the current analysis we choose the strengths of the forces which are "free" parameters of the algorithm according to the density of the network connections. Thus we weigh the strengths of the attractive (repulsive) forces using the ratio of the degrees of the nodes and the total number of nodes in the network by defining $\rho_i=d_i/n$, where d_i is the degree of node i and n is the total number of nodes in the network. For a connectivity matrix C_{ij} composed of zeros and ones, $(C_{ij}-1)=-1$ if the nodes i and j are disconnected or $i=j$ and $(C_{ij}-1)=0$ if nodes i and j are connected. Since the forces are directed along the unit vector $(\vec{r}_i-\vec{r}_j)/|\vec{r}_i-\vec{r}_j|$, we use the positive C_{ij} to generate the attractive forces between connected vertices and the negative $(C_{ij}-1)$ for the repulsive forces between disconnected vertices. Then the strengths S_{ij} in Eq. (6) are then chosen as

$$S_{ij}=\begin{cases}(1-\rho_i)C_{ij}+\rho_i(C_{ij}-1), & i\neq j,\\ 0, & i=j.\end{cases} \tag{14}$$

The two terms in Eq. (14) for $i\neq j$ imply that when the degree of connectivity of the ith node increases, the effect of the attractive forces is reduced while the effect of the repulsive forces increases. Similarly, if the degree of the node is very small, the attractive forces are enhanced while the repulsive forces are diminished. This has the extra benefit that vertices with large degree of connectivity, which are more likely to be highly structural (and belong to cores of communities), are "harder" to attract, while vertices with low degree of connectivity will be easily attracted to other vertices thus helping in the process of community formation on different scales.

The dynamics of the vertices is governed by the forces (from the connectivity matrix) and the vertices displacements vary from vertex to vertex according to their mutual connectivities. Thus, after few evolution steps the new vertex positions correctly represent the cluster structure of the network: the mutual distances between vertices belonging to the same cluster become systematically smaller than the distances between vertices from different clusters. In the extreme ideal case where the clusters are totally disconnected from each other (zero links between different communities), the attractions group the connected nodes as single subnetworks, and the repulsions sharply separate the grouped subnetworks from each other. As the number of links between clusters increases, the attractions between connected members in different clusters can after sufficiently many steps link the subnetworks to form a larger community. The different clusters can merge and collapse to a single point.

In each step of the evolution the DSE algorithm calculates the $n-1$ components of all forces. The number of attractive forces is $\langle d\rangle n/2$. If $\langle d\rangle$ the average degree of nodes in our network is fixed independent of n, then the algorithm using attractions only requires $O(n^2)$ calculations. Introducing repulsive forces leads to higher computational complexity $O(n^3)$. Since the algorithm is equally effective for cluster resolution both with and without repulsive forces, the version with only attractive forces is preferable for the case of large networks.

B. Mutual distances

We use the term "mutual distance" to refer to the spatial separation between any two nodes represented as vertices of the $n-1$ dimensional simplex. The mutual distance between two nodes labeled i and j is given by $d_{ij}=|\vec{r}_i-\vec{r}_j|$. In order to identify the communities, an adequate maximum threshold for the mutual distances must be chosen (in the $n-1$ dimensional space). The threshold is a cutoff value applied to the mutual distances to classify the nodes according to their spatial separation in the $n-1$ dimensional space. More specifically let ε be the value for the threshold, then two nodes with labels i and j belong to the same community if $|\vec{r}_i-\vec{r}_j|<\varepsilon$. Filtering the mutual distances in this way only tells which couples of nodes are close enough to belong to the same community. To build the communities we need to identify which of the couples of nodes filtered belong to the same local neighborhood. We perform the partition of the network as follows:

Let V be a vector with n components, $K=\{Q_1,\ldots,Q_n\}$. Initialize $Q_i=\{i\}$, $V_i=i$; for all $i=1,\ldots,n$;

```
1.  for i=1,...,n-1
2.    for j=i+1,...,n
3.      if d_ij<ε and V_j=V_j
4.        V_j=j, Q_i=Q_i∪{j}, Q_j=φ
5.      end if
6.    end for
7.  end for
8.  Remove all remaining Q_i=φ from K.
```

The routine begins by setting a partition of the networks K made of singlets Q_i containing the labels of the individual nodes. This is the worst case for a partition of the network and it is obtained if no distance between nodes is smaller than the threshold. The routine then merges the singlets to the formed clusters if the distance between one of the members of the cluster and the singlet is smaller than ε. At the end of the routine the class K is the desired partition of the network. Note that the number of calculations needed to partition the network is $n(n-1)/2$ and it only needs to be per-

formed once so no computational complexity is added to the DSE algorithm in this process.

Thus we can define a community as the maximal connected neighborhood of nodes of the $n-1$ dimensional simplex whose maximum mutual distance is bounded by a given maximum value ε.

The success of the community separation in the DSE method lies on the fact that nodes belonging to tightly linked groups approach each other to a distance which is considerably smaller than the average distance between nodes in different communities.

Since this method groups the vertices according to how tightly the nodes are connected to each other, finer substructure detection can be achieved by choosing a lower threshold providing better resolution. At each step of a single run of the algorithm one can apply a set of different thresholds. This provides immediate (spectroscopic) resolution for the cluster and all subcluster structure of the network. Note that thanks to our choice of constant forces which do not fall off with distance, this happens at all scales of mutual distances. Choosing a relatively large threshold of mutual distances (a value chosen from the distribution of mutual distances at a high percentile rank) captures the "big picture" of the network's structure. If the identified communities are treated recursively as subnetworks by the same procedure of applying thresholds chosen from the spectrum of each subnetwork's mutual distances distribution the resolution of the identified structure of the network is improved at different scales thereby identifying possible hierarchical structures.

III. RESULTS AND BENCHMARK

It has become customary to test the efficiency of clustering algorithms on a set of computer generated random networks with a well-defined modular structure [14]. The benchmark networks have 128 nodes, a total of 1024 links, and are composed of four clusters containing 32 nodes each. The nodes are connected with a probability p_{in} (p_{out}) for members of the same community (different communities), in a way such that the average degree of every node in the network is $\langle d\rangle = 16$ (this controls the average number of links z_{out} that each node has with members of other communities). We will use the term "external degree" when referring to z_{out}. As z_{out} increases, the numbers of connections inside each cluster decreases, so that the structure becomes fuzzier and more difficult to identify. We can shuffle the labels of the nodes and apply the DSE algorithm to the shuffled network. To relate the communities detected by the algorithm with the "true" communities with the nonshuffled labels we use the following procedure. Let $\{C_i\}_{i=1,2,3,4}$ be the sets of nodes defining clusters 1, 2, 3, 4, respectively, in the original random network and $\{D_j\}_{j=1,\ldots,m}$ be the clusters detected by the algorithm. The following provides a natural procedure for finding the correspondence between the original clusters and those detected by the algorithm. Let us define $M_{ij}=|C_i \cap D_j|$ to be the number of nodes shared by sets C_i and D_j. We will refer to the matrix M_{ij} as "confusion matrix" since this term is typically used for it.

Consider a random network generated by the procedure mentioned above with an average external degree $z_{\text{out}}=4.92$ (this is the maximal value of z_{out} which still allows a neat separation of the communities still observable with very few steps). We shuffle the locations of the nodes in the connectivity matrix of the network and let the DSE algorithm run for only six steps.

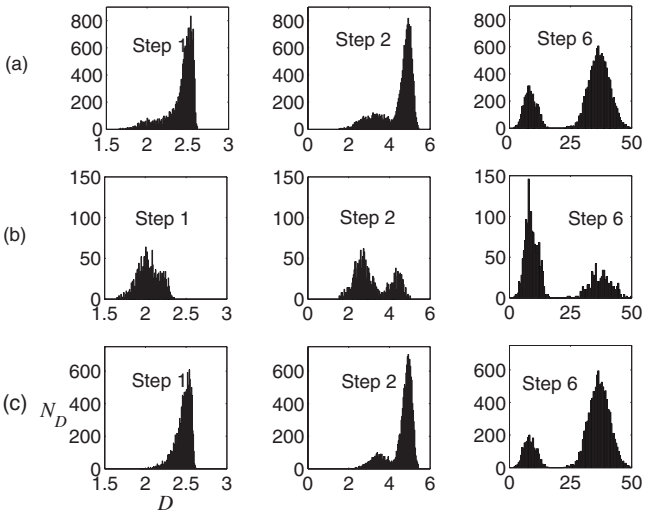

FIG. 1. Histograms of the mutual distances between all pairs of nodes [row (a)], between only connected nodes [row (b)] and between disconnected nodes only [row (c)]. The columns correspond to steps 1, 2, and 6 of the evolution of a generated random matrix with 128 nodes and four clusters, each containing 32 nodes. For each histogram the ordinate axis represents the number of pairs of nodes N_D versus their mutual distances D (abscissa).

Figure 1 shows the histograms of the mutual distances for every pair of nodes in the network at steps 1, 2, and 6 [row 1(a)], for connected nodes only [row 1(b)] and for disconnected nodes only [row 1(c)]. The separation of the communities manifests via the splitting of the single hump shown in step 1 (the first column) into two humps during subsequent steps. Two distinct humps begin to emerge at step number 2 (second column) and the humps are sharply separated at step 6 (third column). For all the histograms where two humps are visible, the left-hand hump corresponds to the mutual distances between nodes inside the same community while the right-hand hump corresponds to the mutual distances between nodes belonging to different communities.

Note that because of the symmetry of the matrix (the clusters have exactly the same sizes and the same densities) we see only two of these humps, but if the symmetry is broken we should expect to see a superposition of humps, each corresponding to a different cluster with a different density and size.

This demonstrates that the structure can be resolved successfully. Step 3 of row 1(b) shows that the maximum distance between nodes belonging to the same community is approximately 17 (in the units for mutual distances), thus, by setting the maximum distance threshold in the surroundings of about 20, one can expect a good identification of the communities. In order to find the threshold systematically for all the simulated networks, we make use of the cumulative distribution of the mutual distances for connected nodes. Figure 2(a) shows the distribution of mutual distances of only con-

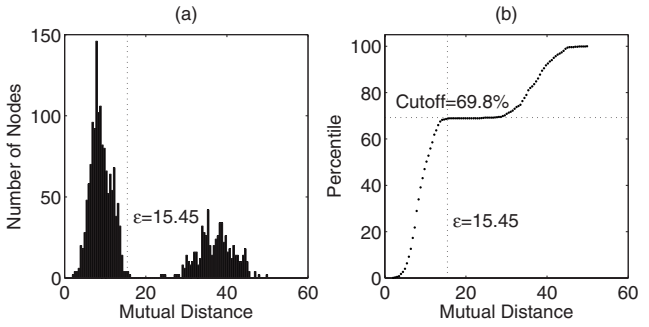

FIG. 2. (a) Histogram of the mutual distances after six steps of the DSE evolution of a simulated matrix with 128 nodes and $z_{out}=4.92$. The threshold $\varepsilon=15.45$ is shown as a vertical dashed line. (b) The cumulative distribution for the mutual distances of the same matrix in (a). The threshold $\varepsilon=15.45$ corresponds to the percentile 69.8 of the distribution (shown as the horizontal dashed line), exactly at the plateau of the cumulative distribution.

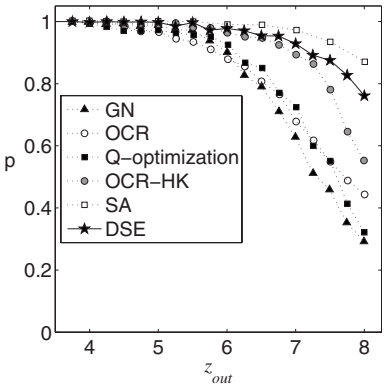

FIG. 3. Fraction p of correctly identified nodes as a function of z_{out} (average number of links between clusters per node) for computer generated random graphs with 128 nodes. The DSE algorithm is represented by stars, for the abbreviations of the other methods see [4].

nected nodes (the same plot in the third step of the second row of Fig. 1). The vertical dashed line shows the threshold $\varepsilon=15.45$. The value of ε can be found observing Fig. 2(a). The horizontal line indicates that the plateau occurring at percentile 69.8 of the mutual distances corresponds to a correct choice of $\varepsilon=15.45$.

The percentage of the number of links between members belonging to the same community compared to the total number of links in the network can be calculated as $\rho=(1-z_{\text{out}}/\langle d\rangle)\times 100\%$. In our example we have $z_{\text{out}}=4.92$ and $\langle d\rangle=16$ so $\rho=69.25$. ρ is in very close proximity to the value 69.8 of the percentile used to find the threshold. This is not a coincidence as all matrices show the same behavior, and it indicates that this procedure for choosing the threshold is accurate, especially for small values of z_{out}. Using the methods described in [2,6] we find for this network a fraction of correctly identified nodes of $p=0.99$ and a normalized mutual information of 0.99, confirming the efficiency of the algorithm and our choice of threshold. These methods are described further in the following section.

On a technical note, the mutual distances are computed along the steps of the simplex evolution as they are required to find the forces between the nodes in the network. Since the threshold is determined exclusively from the distribution of the mutual distances, its calculation does not add any computational complexity to the algorithm.

For the case of a network composed of substructures with different sizes and topologies, the histogram would not be as simple, but rather would reflect the internal structure of all clusters and subclusters; then the hierarchical structures can be identified by means of filtering the mutual distances between connected nodes and by grouping the nodes within adequate ranges of the filtered mutual distances.

In Ref. [6] a criterion known as "fraction of correctly identified nodes" has been proposed to test the efficiency of the community detection algorithms. It essentially looks for the detected community which produces the maximal number of nodes belonging to the original "true" communities. If such community cannot be found, or is undecided which is the maximal set, the nodes are not considered as correctly classified. In terms of the confusion matrix, we can describe the method as follows. The original cluster C_i corresponds to the detected cluster $D_{k(i)}$ for which the number of shared nodes is maximal. If $k(i)$ is not an injective function the nodes associated to C_i are considered incorrectly classified.

Hence, we can measure the efficiency of the detection method by finding the fraction of correctly identified nodes p for each value of z_{out}. The fraction p is then $p=(\Sigma_i M_{ik(i)})/128$.

This benchmark test for the DSE clustering method yields excellent results as shown in Fig. 3 (the abbreviations are the same used in [4]). This figure presents the results of the algorithm test given in the paper [4]. Our results correspond to the line with stars; all other lines correspond exactly to Fig. 1 of paper [4]. (For detailed discussions on the algorithms used to obtain these lines see [4–6], and references therein.) The value of p, corresponding to each value of z_{out} for the DSE algorithm, is obtained after averaging over 50 runs (each run is the sequence of 30 steps).

It can be seen that the values of the fraction of correctly identified nodes for the DSE algorithm are close to the values obtained with the simulated annealing (SA) model, their difference is very small up to a value of $z_{out}=6$. For average external degrees larger than 6.5 the DSE algorithm shows a significantly better performance than the opinion changing rate-Hegselmann and Krause (OCR-HK) method. Considering the fact that the computation required by the SA model can be very demanding [15], this is quite encouraging. The DSE algorithm is thus very efficient in identifying the nodes, and at $z_{\text{out}}=8$ the average fraction of correctly identified nodes is $p>0.75$, which is a great improvement compared to the OCR methods ($p>0.4$ for OCR and $p>0.5$ for OCR-HK). Furthermore, the DSE algorithm provides additional detailed spectroscopic information about internal structures of subclusters in the same run of the algorithm automatically, which is an excellent feature for analysis of networks. It should be noted that it is beyond the scope of this paper to cover all existing algorithms for community detections (for some other approaches see, for example, [16–18], and references therein). We only address the set of algorithms being published with the emphasis of comparisons to other ones

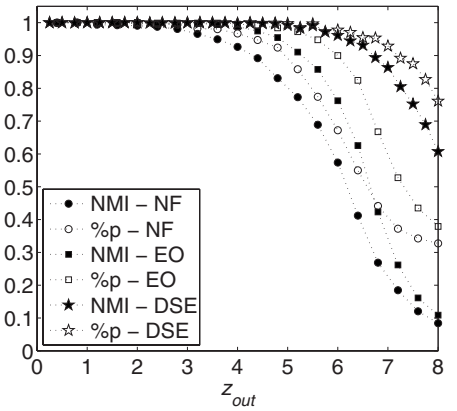

FIG. 4. Benchmark of 30 steps of the dynamical simplex evolution (DSE) versus the fast Newman (FN) and the extremal optimization (EO) algorithms shown in Ref. [2]. We present both methods *%p* (fraction of correctly identified nodes) in hollow symbols and normalized mutual information (NMI) in filled symbols. The DSE algorithm shows very good performance using both criterions and falls down much slower for large external degrees z_{out} than the NF and EO.

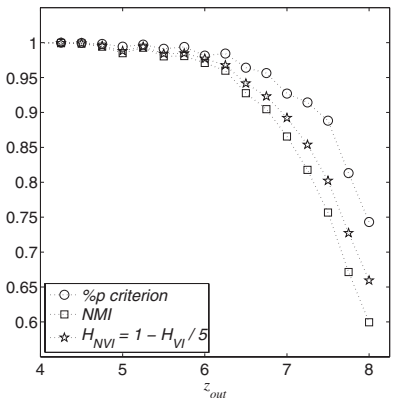

FIG. 5. Three criterion methods for comparing partitions evaluating the performance of the DSE algorithm. The *%p* criterion known as fraction of correctly identified nodes, the normalized mutual information (NMI), and a normalized version of the variation of information (NVI), presented as $H_{\mathrm{NVI}}=1-H_{\text{1-VI/5}}$ (the normalization factor 5 comes from the maximum distance in clustering space between any partition of the *ad hoc* networks and the "true" partition).

using the standard benchmark (considered here) and have been shown to be computationally efficient.

The method of correctly identified nodes serves as a first approach to compare the efficiency of clustering methods. However, as the authors of Ref. [2] note, the p method is questionable when the number of links outside communities z_{out} becomes large as it rewards the identification of smaller clusters found within each of the original communities. The authors of [2] propose instead using the quantity known as normalized mutual information (NMI) H_{NMI} as a more discriminatory measure for this purpose. If the detected partition is exactly the same as the "real" one $H_{\mathrm{NMI}}=1$. For the worst case when the detected partition is totally independent of the real partition (for instance if the algorithm detects only one community, or the nodes are distributed evenly in all communities) the distributions of the real and detected communities are independent, implying $H_{\mathrm{NMI}}=0$.

For the NMI benchmark, we add the results of the DSE algorithm to the ones found in Ref. [2]. This is shown in Fig. 4. The plot shows a comparison of the DSE algorithm versus the fast Newman (FN) and the extremal optimization (EO) methods. The performance of the DSE is very good for large values of the external degree ($z_{out}>5$) compared to the EO.

One should note that the normalized mutual information is symmetric under the exchange of the two partitions. This feature and others mentioned in [2] make the NMI criterion more versatile than the fraction of correctly identified nodes. This symmetry is a consequence of the symmetry observed in the definition of the mutual information, and is exploited further by yet another criterion known as the variation of information (H_{VI}).

The variation of information (H_{VI}) was originally proposed as a criterion for comparing partitions [19]. It can also be used to compare the performance of community detection algorithms by means of using a "distance" on the space of partitions.

As shown in [19], the variation of information is a true metric in the space of all clusterings (partitions). It satisfies the metric axioms of non-negativity, symmetry, and the triangle inequality. Additionally, H_{VI} displays a number of nice features, for instance it depends only on the relative sizes of the clusters, and it is bounded by a maximum value of $\log_2(n)$ where n is the number of nodes in the network. We use base 2 logarithms so that our units are bits. This extreme bounding value happens if one of the partitions consists of all singlets of nodes (i.e., $\{\{1\},\ldots,\{n\}\}$) and the other partition consists of the whole network (i.e., $\{\{1,\ldots,n\}\}$). H_{VI} is thus interpreted as a measure of the distance between two partitions of the same network in the space of all clusterings. The larger the distance H_{VI}, the more different the two partitions are.

If we calculate the "distance" H_{VI} between two arbitrary partitions over a network made of 128 nodes, their maximum possible value of H_{VI} is $\log_2(128)=7$ bits. Nevertheless, the real partition used in the analysis of the 128 nodes *ad hoc* networks is not arbitrary since it is known to be made of four communities each composed of 32 nodes. This tells us that the real partition contains $\log_2(4)=2$ bits of information. Thus for any detected partition of this set of generated networks, the maximum possible variation of information which can be obtained (between the real and the detected partition) is $\log_2(128)-\log_2(4)=5$ bits allowing us to normalize H_{VI} as $H_{\mathrm{NVI}}=1-H_{\mathrm{VI}}/5$ bits. In this way, H_{NVI} is a form of H_{VI} comparable to the criterions of "fraction of correctly identified nodes" and the normalized mutual information within the context of the *ad hoc* networks.

This allows us to present our results for all the three criterion methods in the same plot in the same range as shown in Fig. 5. The H_{NVI} criterion shows larger values than the NMI, this is possibly due to the fact that the NMI method "punishes" some of the configurations indistinctly (for instance, those which yield $H_{\mathrm{NMI}}=0$), while the H_{VI} criterion is more sensitive.

IV. CONCLUSION

In conclusion the general DSE method has several attractive features regarding the purpose of detecting communities in complex networks. First, it does not require discarding (or weakening) connections progressively until a partition is observed, allowing the nodes to interact naturally, utilizing the complete information of the network at every stage of the algorithm. It avoids possibly biasing the results by modifying the network structure. Secondly, it permits the identification of substructures at different scales by setting adequate thresholds in the mutual distances for every step, thereby also adjusting the resolution. For example, for fuzzy structured networks (when the number of connections between communities is large), the resolution in cluster identification can be enhanced by increasing the number of steps and/or decreasing the step sizes. This is extremely important when the network has a self-similar nature, and hierarchical clustering is expected. For our analysis we have used three criterions for comparing partitions, they are the fraction of correctly identified nodes, normalized mutual information (NMI), and variation of information (H_{VI}). Since the H_{VI} criterion possesses all the properties to be a true metric for the space of partitions, and it distinguishes network configurations that the NMI method cannot, we think the H_{VI} method is more appropriate for this type of benchmark between clustering algorithms. Our analysis has shown that the DSE algorithm is very efficient compared to other methods based on its resolution power and moderate computational complexity.

ACKNOWLEDGMENTS

The work of V.G. and V.M. was supported by DARPA through AFRL Grant No. FA8750-04-2-0260, and of Z.N. by the LDRD DR on Physics of Algorithms at LANL.

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).
[2] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech.: Theory Exp. 2005, P09008.
[3] M. E. J. Newman, Eur. Phys. J. B **38**, 321 (2004). M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006).
[4] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, Phys. Rev. E **75**, 045102(R) (2007).
[5] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002); M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[6] M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).
[7] S. Fortunato and M. Barthélemy, Proc. Natl. Acad. Sci. U.S.A. **104**, 36 (2007).
[8] V. Gudkov, J. E. Johnson, and S. Nussinov, e-print arXiv:cond-mat/0209111.
[9] V. Gudkov and S. Nussinov, e-print arXiv:cond-mat/0209112.
[10] G. Behanot, V. Gudkov, S. Nussinov, and Z. Nussinov (unpublished).
[11] V. Gudkov, S. Nussinov, and Z. Nussinov, e-print arXiv:cond-mat/0209419.
[12] S. Nussinov and Z. Nussinov, e-print arXiv:cond-mat/0209155.
[13] V. Gudkov and V. Montealegre, Physica A **387**, 2620 (2008).
[14] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).
[15] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech.: Theory Exp. 2005, P09008.
[16] M. B. Hastings, Phys. Rev. E **74**, 035102(R) (2006).
[17] M. Rosvall and C. T. Bergstrom, e-print arXiv:physics/0612035.
[18] H. N. Djidjev, e-print arXiv:0707.2387.
[19] M. Meila, J. Multivariate Anal. **98**, 873 (2007).