Open access • Book Chapter • DOI:10.1007/978-3-642-04747-3_20

A Comparison of Community Detection Algorithms on Artificial Networks

Günce Keziban Orman, Vincent Labatut

Institutions: Galatasaray University

Published on: 07 Oct 2009 - Discovery Science

Topics: Complex network and Hierarchical network model

A Comparison of Community Detection Algorithms on Artificial Networks

Günce Keziban Orman, Vincent Labatut

A Comparison of Community Detection Algorithms on Artificial Networks

Günce Keziban Orman[1,2] and Vincent Labatut[1]

[1] Galatasaray University, Computer Science Department, Ortaköy/İstanbul, Turkey
[2] TÜBİTAK, Gebze/Kocaeli, Turkey
keziban.orman@uekae.tubitak.gov.tr
vlabatut@gsu.edu.tr

Abstract. Community detection has become a very important part in complex networks analysis. Authors traditionally test their algorithms on a few real or artificial networks. Testing on real networks is necessary, but also limited: the considered real networks are usually small, the actual underlying communities are generally not defined objectively, and it is not possible to control their properties. Generating artificial networks makes it possible to overcome these limitations. Until recently though, most works used variations of the classic Erdős-Rényi random model and consequently suffered from the same flaws, generating networks not realistic enough. In this work, we use Lancichinetti *et al*. model, which is able to generate networks with controlled power-law degree and community distributions, to test some community detection algorithms. We analyze the properties of the generated networks and use the normalized mutual information measure to assess the quality of the results and compare the considered algorithms.

Keywords: Complex networks, Community detection, Algorithms comparison.

1 Introduction

Complex networks are now a popular tool to model a given system, by representing its components and their interactions with nodes and links, respectively. This model can then be analyzed or visualized thanks to some of the many tools designed for graph mining. Complex networks have been used in very different application domains, such as physics, biology, social science or computer networks [1].

Among the various approaches used to study complex networks properties, community detection has become one of the most popular ones. A community, or cluster, is generally defined as a subset of nodes densely interconnected relatively to the rest of the network [2]. Many different community detection algorithms have been defined to identify these subsets. They are generally based on classical clustering principles adapted to graphs, using hierarchical or optimization methods. Hierarchical approaches divide or merge communities by considering the distance or similarity between them, whereas optimization approaches partition the network according to a given criterion.

Authors traditionally test their community detection algorithms on a few real [3-11] or artificial [2-6, 10] networks. Limiting these tests to real networks can be considered as an issue for several reasons. First, building such networks is a costly and difficult task, and determining reference communities can only be done by experts. This leads to small networks, where actual communities are not always defined objectively, or even known. Second, a complex network is characterized by various statistics like its average degree, degree distribution, shortest average path, etc. By definition, it is not possible to control these features in a real network. This means the algorithm is tested on a very specific and limited set of features.

Artificial networks seem to overcome these limitations, because it is possible to randomly generate many of them, while controlling their properties. All that is needed is a generative model able to produce networks with features similar to those of real networks. Of course, artificial networks must not be seen as a substitute to real networks, but rather as a complement. In the context of testing community detection algorithms, the most popular generative model is the one defined by Newman and Girvan [4], which is used in all the works cited above. It is a variation of the classic Erdős-Rényi random model [12] (or Poisson random model), and it consequently suffers from the same limitation: the generated networks do not show a realistic topology [13, 14].

Some recent works tried to improve this by defining more realistic models, able to mimic some of the real networks features. In this work, we use the model proposed by Lancichinetti *et al.* [14], which is able to generate networks with controlled power-law degree and community distributions. Our purpose is to generate a set of artificial networks with various size and properties, and to use it to test the existing community detection algorithms. We use the normalized mutual information measure [15-17] to assess the quality of the results and compare the considered algorithms.

In section 2, we explain what the properties of a complex network are. It is of course an open question to decide how a complex network can be described by a few features, but we kept only the most widely used ones. In section 3, we focus on the community detection task. We first describe its general mechanisms, and the modularity measure, which is used as an optimization criterion in many algorithms. Then, we list the algorithms we chose to compare. In section 4, we explain how we generated our test set of artificial networks, and we give some explanation about the normalized mutual information measure we used to assess the algorithms performance. In section 5, we present and discuss our results, focusing first on the observed properties of the generated networks, and then on the comparison of the algorithms' performances.

2 Complex Networks Properties

Undirected real networks are known to share some common properties. In this section, we present the most prominent ones: small-worldness, transitivity, degree-related properties and community structure. Many other properties can be used to describe a network, either by analyzing some measure, like betweenness-centrality distribution [18] or network diameter [19], or by counting the number of occurrences

of a given substructure like motifs in [20]. But their use is not really widespread, and we would consequently lack experimental values to exploit them in this work.

Small-World. A network is said to have the small-world property if, for a fixed average degree, the average distance (i.e. the length of the shortest path) between pairs of nodes increases logarithmically with the number of nodes n [1]. This property can be interpreted as propagation efficiency: spreading on the network remains relatively fast even if the network grows.

Transitivity. The transitivity property is measured by a transitivity coefficient, also called clustering coefficient [21]. Different versions of this coefficient exist, but they all try to assess the density of triangles in a network. The higher this coefficient, the more probable it is to observe a link between two nodes which are both connected to a third one. Independently of the considered coefficient version, a real network is supposed to have a higher transitivity than a Poisson random network (such as those generated by the Erdős–Rényi model [12]) with the same number of nodes and links, by a factor of order n [1].

Degree. Networks can also be described according to their degree distribution. In most real networks, this distribution follows either a power or an exponential law. In other terms, the probability for a node to have a degree k is either $p_k \sim k^{-\gamma}$ or $p_k \sim e^{-k/\kappa}$ [1]. Networks with a power-law degree distribution are the most common. They are called scale-free, because their degree distribution does not depend on their size (some other properties may, though). Experimental studies showed that the γ coefficient usually ranges from 2 to 3 [1, 19, 22]. It is known that for values of γ smaller than 3.48, there is a high probability the network contains one giant component and several small ones (a component is a separated subgraph), or even only one component (the network being completely connected) [22].

In a real network, the average and maximal degrees generally depend on the number of nodes it contains. For a scale-free network, it is estimated to be $\langle k \rangle \sim k_{max}^{-\gamma+2}$ [19, 22] and $k_{max} \sim n^{1/(\gamma-1)}$ [1], respectively.

The degree correlation of a network constitutes another interesting property. The question is to know how a node degree is related to its neighbors'. Real networks usually show a non-zero degree correlation. If it is positive, the network is said to have assortatively mixed degrees, whereas if it is negative, it is disassortatively mixed [1]. According to Newman, social networks tend to be assortatively mixed, while other kinds of networks are generally disassortatively mixed. Nodes with high degree are called hubs, because they have a more central position in the network.

Community. In this work, our focus is on detecting communities in networks. Of course, it is important to note that not all real networks have a community structure. According to Newman though, it is a common feature in biological and social networks [1]. When the community structure is present, the community size distribution seems to follow a power-law distribution [23] with a parameter β ranging from 1 to 2 [5, 24].

3 Community Detection

Complex networks have been used widely to model real-world systems in many application fields. When analyzing a complex network, the problem of identifying its communities is universal, and has consequently been raised in many domains, leading to different solutions. Many of them rely on Newman's modularity to assess the quality of their results, so we will first introduce this measure. Then, we will present the principles of community detection, and give a short description of the algorithms we chose to compare.

3.1 Modularity

The modularity measure has been presented by Newman and Girvan [2] to assess the quality of a network partition. They first define what could be called a community contingency matrix, whose elements p_{ij} represent the fraction of total links from a node in community i towards a node in community j. The fraction of links inside community i is therefore p_{ii}. Moreover, since we are considering undirected networks, we have $p_{ij} = p_{ji}$ and the matrix is symmetric.

Let p_{i+} and p_{+j} be the sums over row i and column j, respectively. If the network has no community structure, or if the considered communities are not defined accordingly to the network structure, then one can suppose the links are randomly distributed. Under this hypothesis, the expected fraction of links inside community i can be estimated as the probability for a link to start in community i, which is p_{i+}, multiplied by the probability to end in community i, which is p_{+i}. The matrix being symmetric, we have $p_{i+}p_{+i} = (p_{i+})^2$. The modularity measure is defined as the difference between the observed and expected fractions of links in each community, summed over all communities:

$$Q = \sum_i p_{ii} - \sum_i (p_{i+})^2 \quad (1)$$

When the communities are not better than a random partition, or when the network does not exhibit any community structure, Q is negative or zero. Its superior limit is 1, but it can be approached only if the network has a strong community structure and if the communities have been perfectly detected.

Interestingly enough, the modularity measure is similar to the numerator of chance-corrected measures used to assess the performance of classic classifiers, such as Cohen's κ coefficient [25]. The general formula for these measures is $(P_o - P_e)/(1 - P_e)$, where P_o is the observed agreement and P_e is the expected agreement between the classifier results and the classified data. But unlike modularity, chance corrected measures are normalized by the dividing term $(1 - P_e)$, which represents a perfect classifier result (reaching a 1 observed agreement). Of course, it is not possible to process the corresponding value in the case of modularity, because the superior limit for $\sum_i p_{ii}$ depends on the community structure of the network, and is usually less than one (whereas 1 is an absolute value for classic classifiers).

The modularity measure is known to have some flaws. For example, it is sensitive to community size [26] and it is possible to find partitions of Poisson random networks with relatively high modularity values [27] (although they have no community strucuture). However, many community detection algorithms use it as an optimization criterion, as we will see in the following section.

3.2 Algorithms for Community Detection

It is difficult to categorize the community detection algorithms, but one could group them in three different families: hierarchical, optimization, and others.

Early solutions are based on hierarchical approaches whose result is a tree of communities called dendrogram. Agglomerative approaches starts with as many communities as nodes, each node having its own community, and iteratively merge these communities until only one giant community remains. On the opposite, divisive approaches start with one community containing all nodes, and iteratively split the communities until each node constitute one community. The communities to be merged or split are chosen accordingly to some distance or similarity function which allows detecting which communities are similar (agglomerative approach) or heterogeneous (divisive approach). What distinguishes algorithms in this family is mainly the nature of the distance or similarity function. The result being a dendrogram, one still needs to find out where to perform a cut in order to get an actual partition. For instance, one can compute the modularity at each level, and use the partition with maximal modularity.

The optimization-based approaches use a measure to estimate the quality of a network partition. This measure is, most of the time, Newman's modularity [2]. The general algorithm consists in first processing several partitions of the network (randomly or by following a fitting function) and second keeping the best one according to the quality measure. This partition can then be refined in order to get a better quality. Modularity is a costly measure to process, hence the numerous algorithms defined for its optimization [3, 6, 24].

The last family contains all the remaining approaches. Some use different principles coming from classical clustering like density-based clustering [7]; some are agent-based [8]; some allows finding overlapping communities (one node can be a part of several communities at once) [28]; some use a latent space approach to process the probability for a node to belong to a community [9].

This work consists in comparing community detection algorithms on many generated networks, so we chose to focus first on the following algorithms, which are fast and simple.

Fast Greedy Algorithm. This algorithm was developed by Newman *et al.* [10, 24]. It is modularity-based and uses a hierarchical agglomerative approach. It is called fast greedy because thanks to a standard greedy method, it is significantly faster than older algorithms.

Walktrap Algorithm. This algorithm by Pons and Latapy [29] uses a hierarchical agglomerative method. Here, the distance between two nodes is defined in terms of random walk process. The basic idea is that if two nodes are in the same community, the probability to get to a third node k located in the same community through a random walk should not be very different for i and j. The distance is constructed by summing these differences over all nodes, with a correction for degree.

Eigenvector Algorithm. This algorithm by Newman [30] is modularity-based, and it uses an optimization method inspired by graph partitioning techniques. It relies on the eigenvectors of a so-called modularity matrix, instead of the graph Laplacian traditionally used in graph partitioning.

Label Propagation Algorithm. This algorithm by Raghavan *et al.* [11] uses the concept of node neighborhood and the diffusion of information in the network to identify communities. Initially, each node is labeled with a unique value. Then an iterative process takes place, where each node takes the label which is the most spread in its neighborhood. This process goes on until one of several conditions is met, for instance no label change. The resulting communities are defined by the last label values.

Spinglass Algorithm. This algorithm by Reichardt and Bornholdt [31] is an optimization method relying on an analogy between the statistical mechanics of complex networks and physical spin glass models.

4 Method

In order to compare the selected algorithms, we chose to generate a set of artificial networks. If we want the results to hold when the algorithms are applied on real networks, our artificial networks properties must be the most similar possible to those we previously described for real networks. Another important point is the assessment of the results quality, which must be reliable in order to compare efficiently the communities detected by the tested algorithms. In this section, we present the model we used to generate our test data and the measure we chose to assess the algorithms performance.

4.1 Network Generation

In many community detection works [3, 32, 33], artificial community-structured networks are generated with models similar to the one defined by Newman and Girvan [4, 10]. It relies on the principle of the Erdős-Rényi model [12]: each community corresponds to a Poisson random network, with a probability p_{in} to have a link between two of its nodes (an internal link). Another probability p_{out} is used to add links between nodes from different communities (external links). The

probabilities are constrained so that $p_{in} > p_{out}$ and the average degree d of the resulting network tends towards a fixed value.

This model lacks some of the properties we described earlier: the degree distribution and the community size distribution do not follow a power-law, and we have no information about the other properties. For this reason, we chose to use a more recent model defined by Lancichinetti *et al.* [14] to generate our test set of artificial networks. It allows generating random networks with a community structure and a power-law degree distribution. Moreover, the size of the resulting communities also follows a power-law distribution.

This method needs the following compulsory parameters: the number of nodes n, the desired average $\langle k \rangle$ and maximum k_{max} degrees, the exponent γ for the degree distribution, the exponent β for the community size distribution, and a value μ called the mixing coefficient. The latter represents the average proportion of links between a node and nodes located outside its community, $1 - \mu$ being the proportion of links with nodes located in the same community. This leads to the concepts of internal and external degrees, corresponding to the number of links a node has inside and outside its community, respectively. For a node of degree k, we then have the values $(1 - \mu)k$ for the internal degree and μk for the external degree. Of course, these values hold in average, but can only be approximated when considering a given node. Two additional parameters, the minimum and maximum community sizes, can also be optionally precised. If this is not the case, they are automatically set to values smaller than the minimal degree and greater than the maximal degree, respectively. This way, every node can fit in a community, whatever its degree.

The generation is performed in three steps. First, the well-known configuration model [34] is used to generate a scale-free network corresponding to the specified γ parameter. Second, the community sizes are drawn in accordance with the β parameter, and each node is randomly affected to a compatible community. Compatible means here that the community size must be greater or equal to the node internal degree. Some specific mechanisms ensure the convergence of the processing, see [14] for more details. Third, some links are rewired in order to respect the mixing coefficient. For a given node, the total degree is not modified, but the ratio of internal and external links is changed so that the resulting proportion gets close to μ.

Our goal was to compare the performance of community detection algorithms, so we generated networks with parameters consistent with what is observed in real community-structured networks. We used the value 1000 for the number of nodes n. The β and γ exponents ranged from 1 to 2 and from 2 to 3, respectively. We used values of μ in $[0.05; 0.95]$ with a 0.05 step. For each set of parameters, we generated 25 networks in order to deal with possible discrepancies in the networks properties due to the random generation.

In rare occasions, we observed that some parameters can cause several components to appear in the same network. Some algorithms like Walktrap cannot be applied on such networks, so we decided to randomly connect these components in order to be able to apply all the algorithms.

4.2 Performance Assessment

As we stated before, the modularity measure is a standard for assessing the quality of a network partition. But it was designed to be an approximation of the partition quality, to guide community detection algorithms when the actual communities are unknown. The value computed for a given situation depends on both the quality of the detected communities and of the nature of the network community structure.

This dependence to the network structure prevents from using modularity to compare algorithm performances on different networks. Furthermore, we will use artificial networks, whose communities are known *a priori*. In this context, modularity is not an appropriate measure, because it does not make use of this important information. For interpretation purposes, we nevertheless processed the modularity for the various tests we performed (several tested algorithms use modularity during their processing).

Instead of modularity, we used the normalized mutual information measure (NMI). It was defined in the context of classical clustering to compare two different partitions of one data set [15, 16]. It was shown to be an efficient way to assess the quality of estimated network communities by Danon *et al.* [17].

The measure is derived from a confusion matrix whose element m_{ij} represents the number of nodes put in community i by the considered algorithm, whereas they actually belong to community j. This matrix is usually rectangular, because the algorithm does not necessary find the correct number of communities.

$$I = \frac{-2\sum_i \sum_j m_{ij} \log\left(n m_{ij} / m_{i+} m_{+j}\right)}{\sum_i m_{i+} \log(m_{i+}/n) + \sum_j m_{+j} \log\left(m_{+j}/n\right)} \quad (2)$$

If the estimated communities correspond perfectly to reality, the measure takes the value 1, whereas it is 0 when the estimated communities are independent from the actual ones.

5 Results and Discussion

5.1 Generated Networks

The model from Lancichinetti *et al*. [14] allows controlling most of the network properties: number of nodes, degree distribution, maximal and average degrees and community size distribution. For these properties, we used realistic values in accordance with the literature (cf. the Complex Network Properties section). The question is to know whether the uncontrolled properties (average distance, transitivity, correlation degree), arising from the processing, are realistic too. Furthermore, we would like to know if and how changes in the controlled parameters affect the uncontrolled ones. This is an important matter, because such a change may influence the algorithms' performances, which could therefore be explained either by a direct or an indirect effect. By direct effect, we mean the observed performance modifications are related to the changed controlled properties. By indirect effect, we

mean they are related to a change in some uncontrolled properties, caused itself by the change in controlled properties.

In the following, we will discuss separately the relation between each parameter and the uncontrolled properties. The numbers indicated in parenthesis correspond to the processed (Pearson's) correlation values between the considered parameter and uncontrolled property, for 1000 nodes networks.

The variations in the average and maximal degrees have little or no effect on the degree correlation and transitivity coefficient (-0.14 and 0.09, respectively), but there is a direct relation with the average distance (-0.66). Unsurprisingly, it decreases dramatically when the average degree increases, certainly due to the rise in the number of links.

The β parameter has little or no effect on the average path length (0.01) and the transitivity coefficient (0.05), but it relatively affects the degree correlation (0.37). The β parameter controls the homogeneity of the community sizes: when it increases, the communities tend to be more uniform in terms of size [14]. Our interpretation is that with a small beta, we have many small communities with no hubs, much less medium communities with a few hubs, and a few big communities with more hubs. Medium community hubs have less chance to get linked with other hubs, because there are only a few hubs in their community, and links between communities are rarer, which prevents them to get linked with hubs in other communities. When beta increases, this chance also increases because the number of hubs in the same community gets larger.

The γ parameter has little or no effect on average distance (0.07) and transitivity (-0.06), but it relatively affects the degree correlation (-0.26). When γ increases, the network degree distribution becomes more homogeneous, so this is consistent with the fact that degree correlation is close to zero in Poisson random networks.

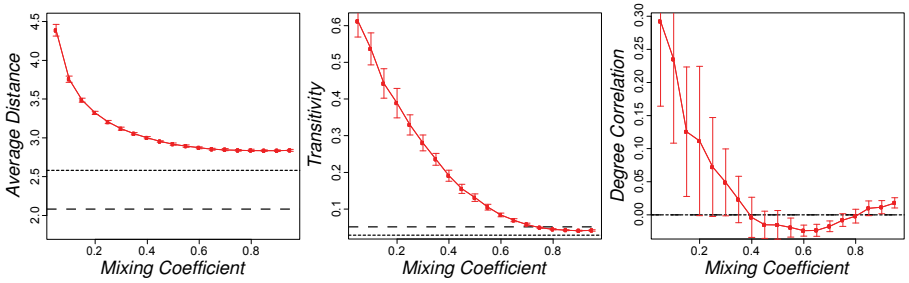

Fig. 1. Influence of the mixing coefficient μ on the properties of the generated networks. The controlled parameters are $n = 1000$, $\langle k \rangle = 30$, $k_{max} = 90$, $\beta = 2$ and $\gamma = 3$. Each point corresponds to an average over 25 generated networks. The dotted and dashed horizontal lines represent the expected values for the same properties in networks generated with the configuration model [34] and Poisson model [12], respectively, using similar parameters.

The most influent parameter is the mixing coefficient μ, as shown in Fig. 1. The computed correlations are not necessarily high, but the plots show a non-linear relationship between μ and all three uncontrolled properties. As shown on the plot, the average distance decreases when μ increases. However, we performed additional measurements on networks with sizes between 100 and 100000 nodes, and observed a clear logarithmic relationship between the size and the average distance, which is

consistent with real networks features. The transitivity is very high for low μ values, but gets down to the level of Poisson random networks when μ reaches 0.7. In the same way, for low μ values, the degree correlation is relatively high, but quickly decreases until μ reaches 0.4 or 0.5, and then stays close to zero. Interestingly, $\mu = 0.5$ corresponds to a limit above which the proportion of external links is higher than the proportion of internal links. In other terms, when μ goes above this limit, the communities are not well defined anymore, and we have a scale-free network with no community structure. Here, we must recall Lancichinetti *et al*. method consists in using the configuration model to generate a scale-free network, which is then partially rewired to create a community structure. For a given node, there are usually many more nodes outside than inside its community. Therefore, the higher μ and the lesser the original network is modified. Put differently: when μ grows, the generated networks get more similar to scale-free networks generated by the configuration model. The configuration model is known to produce networks with no degree correlation [35]. Furthermore, Newman [1] showed that when it is used to generate scale-free networks, for $\gamma > 7/3$ the transitivity tends toward zero as the number of nodes is increasing. Our measures show close to zero degree correlation and transitivity when μ gets close to 1, which is consistent with the previous remarks. The average distance is also close to what is expected from a configuration model-produced network [36]. Using smaller μ values, i.e. defining more distinct communities, makes all three properties grow. The effect on degree correlation could be due to the apparition of hub-to-hub links between communities. The definition of community used here relies on stronger inner density, and is therefore related to the concept of transitivity, which may explain its increase. The disappearance of shortcut links between the communities could explain the observed decrease in average distance.

To conclude these observations, we can state the generated networks show some reasonably realistic properties when μ is relatively small. However, increasing this parameter not only causes communities to become less distinct, but also makes the whole network becoming less realistic, its average distance, transitivity and degree correlation decreasing rapidly.

5.2 Algorithms Performance

The results from the five algorithms are presented in Fig. 2. We can distinguish three kinds of results: Spinglass and Walktrap perform generally very well; Label Propagation also performs well, but is more sensitive to decreases in μ; Eigenvector and Fast greedy are clearly below the others, especially for networks with high degrees. More generally, all the algorithms are sensitive to changes in the average and maximal degree, and have better performances when it increases, as Lancichinetti *et al.* previously noticed on different algorithms [14]. But this sensibility is not the same for all of them, as we can observe different decreases in performance when μ is increasing. This general sensitivity to μ is not surprising, since an increase in μ means the communities are vanishing. Spinglass and Walktrap are the most robust, with NMI results remaining at 1 until they suddenly drop between $\mu = 0.6$ and 0.8 for the

two higher values of $\langle k \rangle$ and k_{max} (last two rows). For the lower degrees values (first row), the decrease is more regular and starts from $\mu = 0.05$.

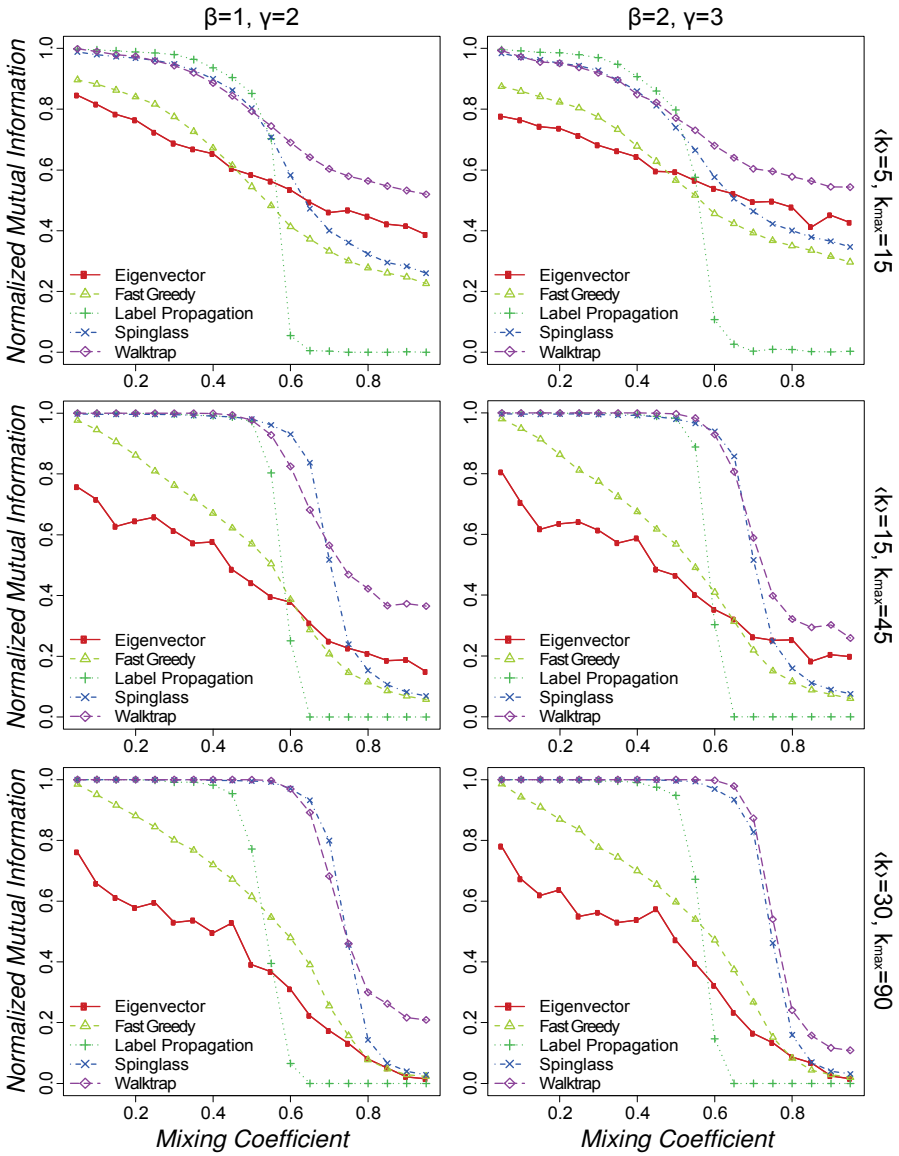

Fig. 2. Comparison of the five algorithms results for $n = 1000$. On the left column: $\beta = 1$ and $\gamma = 2$, on the right column $\beta = 2$ and $\gamma = 3$. On the first row $\langle k \rangle = 5$ and $k_{max} = 15$, on the second one $\langle k \rangle = 15$ and $k_{max} = 45$, and on the third one $\langle k \rangle = 30$ and $k_{max} = 90$. Each point corresponds to an average over 25 generated networks.

For Eigenvector and Fast greedy, the performance drop takes place sooner, and is almost linear starting from $\mu = 0.05$, with all three tested degree values. Label

Propagation behavior is apart: it performs almost as well as Spinglass and Walktrap, but its performance drop happens sooner, between $\mu = 0.5$ and 0.6, and is more sudden.

When observing the joint effect of the mixing coefficient and the average and maximum degrees on the performance, it is interesting to observe the reversal taking place around $\mu = 0.75$, as illustrated by Fig. 3 for Walktrap and Spinglass. Below this limit, the higher the degree and the better the performance. Above this limit, the lower the degree and the better the performance. This means high density helps discovering community structure when it is strong, whereas it hides it when it is weak. But as we stated in the previous section, the generated networks become less realistic when μ increases, so the observed change in performance could actually be caused not directly by the degree variations, but by consequent decreases in the transitivity or degree correlation.

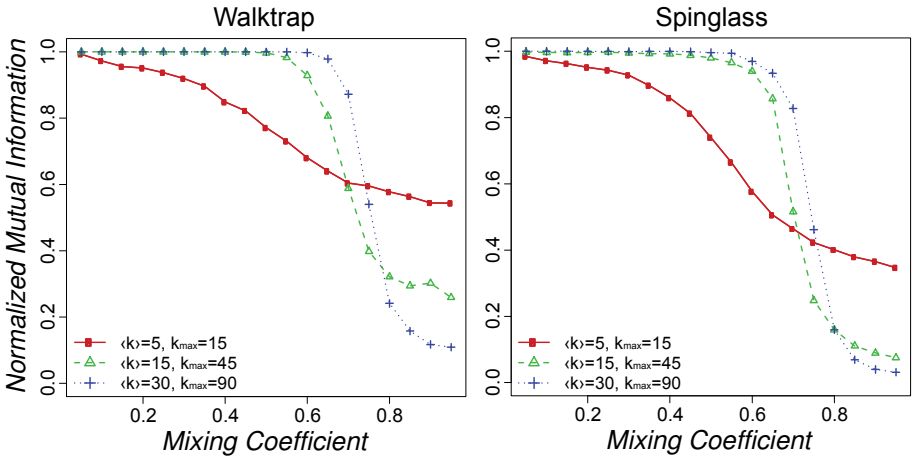

Fig. 3. Joint influence of the mixing coefficient μ and the average degree $\langle k \rangle$ on the performance of two algorithms: Walktrap on the left and Spinglass on the right. Each point corresponds to an average over 25 generated networks, with $\beta = 2$ and $\gamma = 3$.

The β and γ parameters do not seem to affect any of the algorithms (correlation smaller than 0.06 for all five), except for Walktrap, for which it looks like γ has an effect similar to the degree effect observed before. In other terms, the scale-free property makes it easier for Walktrap to discover communities when the community structure is strong, but makes it more difficult when it is weak.

6 Conclusion

In this paper, we compared five different community detection algorithms. We used a set of artificial networks generated with the model defined by Lancichinetti *et al.* [14], which allows randomly producing networks with a community structure and power-law degree and community size distributions. To our knowledge, this type of comparative study was never conducted on such realistic networks before. We used

the normalized mutual information measure [15-17] to assess the performance of the algorithms. Our results show that among the Fast Greedy [10, 24], Walktrap [29], Eigenvector [30], Label Propagation [11] and Spinglass [31] algorithms, Walktrap and Spinglass get generally the best results. They succeed in identifying the communities even for high mixing coefficient values. Label Propagation has also excellent results, but its performance drop happens before Spinglass and Walktrap. Fast greedy and Eigenvector are clearly outclassed by all three other algorithms.

After having analyzed the data, we concluded the mixing coefficient and average and maximum degrees have a strong joint effect on the algorithms results. A higher density tends to improve community finding when the communities are distinct, but makes it harder to find them when the community structure is weak. In these algorithms and in the modularity measure, a community is defined as a subset of nodes densely interconnected relatively to the rest of the network. This definition does not hold anymore when $\mu > 0.5$, which means above this limit, the network structure does not reflect the community structure. In other terms, the information conveyed by the network links is not pertinent anymore, and this can explain the observed joint effect. Moreover, increases in the mixing coefficient also make the networks becoming less realistic, which could as well be a cause for the observed drop in performance.

Our work can be seen as a first attempt at comparing community detection algorithms, and can be extended in several ways. The generative model we used is more realistic than earlier ones, but we observed that increasing the mixing coefficient μ makes the produced networks less realistic (strong decrease in the average distance, degree correlation and transitivity). We suppose this is due to the use of the configuration model [34] by Lancichinetti *et al*. [14] to produce an initial network, which is then modified to create the community structure. Maybe this could be corrected by using another model instead, such as preferential attachment [37] (or one of its variations), able to generate networks with more realistic properties. Of course there is no certainty about whether or not these properties would resist the modifications performed on the initial network.

We only considered a few properties to analyze the artificially generated networks, and some additional properties, maybe more community-oriented (see [19]) could be used to have a more precise idea of their realism. Moreover, real networks properties are usually described commonly, but there may be strong differences between the various types of real networks such as social networks, biological networks, information networks, etc. [1]. In that case, a proper test should compare algorithms on different types of corresponding artificial networks.

We compared the algorithms on networks containing only 1000 nodes. Real networks are generally much bigger, in the order of tens of thousands or millions of nodes. For more significance, the algorithms should be tested on this type of networks, but this raises two problems: 1) processing community detection on such huge networks is significantly more time expensive, and 2) determining realistic average and maximal degrees is difficult because of the heterogeneity observed in real networks for these properties. The second point is important, since we observed the performance of a given algorithm could vary strongly in function of these sole properties. We also limited this comparison to the fastest algorithms, again for

computability and time reasons. A proper exhaustive test should consider more expensive algorithms (see [17, 19]).

Finally, we sometimes observed extreme disagreements between the final modularity measure processed by the various algorithms and the information measure corresponding to their performance. It should be interesting to process the optimal modularity over all the tested algorithms and to study how it evolves relatively to the networks properties and the measured performances.

Acknowledgments. The authors would like to thank the TÜBİTAK (Scientific and Technological Research Council of Turkey) for its support, and the anonymous referees for their valuable comments

7 References

1. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167-256
2. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69** (2004) 026113
3. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Physical Review E **72** (2005) 027104
4. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America **99** (2002) 7821-7826
5. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435** (2005) 814-818
6. Reichardt, J., Bornholdt, S.: Detecting Fuzzy Community Structures in Complex Networks with a Potts Model. Physical Review Letters **93** (2004) 218701
7. Falkowski, T., Barth, A., Spiliopoulou, M.: DENGRAPH: A Density-based Community Detection Algorithm. IEEE/WIC/ACM International Conference on Web Intelligence (2007) 112-115
8. Liu, Y., Wang, Q., Wang, Q., Yao, Q., Liu, Y.: Email Community Detection Using Artificial Ant Colony Clustering. Advances in Web and Network Technologies, and Information Management. Springer, Berlin / Heidelberg (2007) 287-298
9. Hoff, P., Raftery, A., Handcock, M.: Latent space approaches to social network analysis. Journal of the American Statistical Association **97** (2002) 1090-1098
10. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E **69** (2004) 066133
11. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76** (2007) 036106
12. Erdos, P., Renyi, A.: On random graphs. Publicationes Mathematicae **6** (1959) 290-297
13. Danon, L., Diaz-Guilera, A., Arenas, A.: The effect of size heterogeneity on community identification in complex networks. Journal of Statistical Mechanics-Theory and Experiment (2006) 11010
14. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys Rev E Stat Nonlin Soft Matter Phys **78** (2008) 046110
15. Kuncheva, L.I., Hadjitodorov, S.T.: Using diversity in cluster ensembles. IEEE International Conference on Systems, Man and Cybernetics, Vol. 2 (2004) 1214 - 1219
16. Fred, A.L.N., Jain, A.K.: Robust Data Clustering. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2. IEEE Computer Society (2003) 128-136

17. Danon, L., Díaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. J. Stat. Mech. (2005) P09008
18. Holme, P., Kim, B.J., Yoon, C.N., Han, S.K.: Attack vulnerability of complex networks. Physical Review E **65** (2002) 026139
19. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.: Complex networks: structure and dynamics. Physics Reports **424** (2006) 175-308
20. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon1, U.: Network Motifs: Simple Building Blocks of Complex Networks. Science **298** (2002) 824 - 827
21. Watts, D., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393** (1998) 409-410
22. Barabasi, A., Albert, R.: Statistical mechanics of complex networks. Reviews of Modern physics **74** (2002) 47-96
23. Guimerà, R., Danon, L., Díaz-Guilera, A., Giralt, F., Arenas, A.: Self-similar community structure in a network of human interactions. Physical Review E **68** (2003) 065103
24. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E **70** (2004) 066111
25. Cohen, J.: A Coefficient of Agreement for Nominal Scales. Educational Psychology Measurement **20** (1960) 37-46
26. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. Proceedings of the National Academy of Science of the USA **104** (2007) 36-41
27. Guimerà, R., Sales-Pardo, M., Amaral, L.A.N.: Modularity from fluctuations in random graphs and complex networks. Physical Review E **70** (2004) 025101
28. Derenyi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. Physical Review Letters **94** (2005)
29. Pons, P., Latapy, M.: Computing communities in large networks using random walks. Computer and Information Sciences - Iscis 2005, Proceedings **3733** (2005) 284-293
30. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74** (2006) 036104
31. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Phys. Rev. E **74** (2006) 016110
32. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the United States of America **101** (2004) 2658-2663
33. Donetti, L., Munoz, M.A.: Detecting network communities: a new systematic and efficient algorithm. Journal of Statistical Mechanics: Theory and Experiment (2004) P10012
34. Molloy, M., Reed, B.: A critical point for random graphs with a given degree sequence. Random Structures and Algorithms **6** (1995) 161-179
35. Serrano, M., Boguñá, M.: Weighted Configuration Model. AIP Conference Proceedings **776** (2005) 101
36. Chung, F., Lu, L.: The average distances in random graphs with given expected degrees. PNAS **99** (2002) 15879-15882
37. Barabási, A.-L., Albert, R.: Emergence of Scaling in Random Networks. Science **286** (1999) 509-512