# Put Three and Three Together: Triangle Driven Community Detection

ARNAU PRAT-PÉREZ, DAMA-UPC, Universitat Politècnica de Catalunya
DAVID DOMINGUEZ-SAL, Sparsity Technologies
JOSEP-M. BRUNAT, Departament Matemàtica Aplicada II, Universitat Politècnica de Catalunya
JOSEP-LLUIS LARRIBA-PEY, DAMA-UPC, Universitat Politècnica de Catalunya

Community detection has arisen as one of the most relevant topics in the field of graph data mining due to its applications in many fields such as biology, social networks or network traffic analysis. Although the existing metrics used to quantify the quality of a community work well in general, under some circumstances they fail at correctly capturing such notion. The main reason is that these metrics consider the internal community edges as a set, but ignore how these actually connect the vertices of the community. We propose the Weighted Community Clustering ($WCC$), which is a new community metric that takes the triangle instead of the edge as the minimal structural motif indicating the presence of a strong relation in a graph. We theoretically analyse $WCC$ in depth and formally prove, by means of a set of properties, that the maximization of $WCC$ guarantees communities with cohesion and structure. In addition, we propose *Scalable Community Detection (SCD)*, a community detection algorithm based on $WCC$, which is designed to be fast and scalable on SMP machines, showing experimentally that $WCC$ correctly captures the concept of community in social networks using real datasets. Finally, using ground truth data, we show that $SCD$ provides better quality than the best disjoint community detection algorithms of the state of the art while performing faster.

## 1. INTRODUCTION

Communities are informally defined as sets of vertices which are internally connected but scarcely connected to the rest of the graph. The retrieval of vertex communities (or clusters) provides information about the sets of vertices that respond to a similar concept [Girvan and Newman 2002]. For instance, in social networks, communities identify groups of users with similar interests, locations, friends or occupations. This information is useful to perform more focused marketing campaigns [Wang et al. 2009], to craft new visual representations of data [Di Giacomo et al. 2007], increasing data locality thanks to a more coalesced data placement [Prat-Pérez et al. 2011], finding expansion terms in query engines [Guisado-Gámez et al. 2014] or for item suggestion [Sozio and Gionis 2010].

Detecting communities is inherently complex, since there is not a consensus on what a community formally is. In the literature, communities are typically defined by means of metrics that quantify how dense or isolated are [Newman and Girvan 2004; Kannan et al. 2004; Leskovec et al. 2010]. Among those metrics, modularity and conductance are those which have become more popular [Fortunato 2010] and precise [Leskovec et al. 2010], respectively. Although these community metrics work well in general, they have problems to correctly quantify the quality of a community under certain circumstances. The main reason for this failure is that they consider all the edges as equally important, without taking into account if they form structures internally. Then, algorithms based on maximizing these metrics end up finding sets of vertices that visually manifest a lack of community structure [Leskovec et al. 2010].

As a first contribution, we propose a new metric called Weighted Community Clustering ($WCC$). $WCC$ takes the triangle instead of the edge as the basic indicator of a strong relation in the graph. A triangle is a transitive relation between three vertices. Social networks are known to contain more triangles than expected in a random graph [Newman and Park 2003; Newman 2001; Shi et al. 2007; Satuluri et al. 2011]. Such a large presence of triangles is a direct consequence of what is known as the “homophily principle” [McPherson et al. 2001], which suggests that similar entities in a network tend to establish connections, such as people with similar interests, members of the same family or work mates. This creates homophilic regions in the graph (the communities) with a larger edge density, and hence with a larger number of triangles. The usage of triangles allows us to differentiate from those casual edges and sets a minimal structural bound, which combined with a proper metric design, makes $WCC$ sensitive to the internal edge layout of the community.

By observing the degenerate behavior of existing community metrics under certain circumstances, we identify a set of basic properties worth to consider when designing a community detection metric, to guarantee communities with a minimum level of cohesion and structure. These properties have been the backbone around which $WCC$ has been shaped. Therefore, as a second contribution, we propose four basic properties and we prove mathematically that $WCC$ is able to correctly capture the concept of a community by fulfilling them. Our approach goes one step further from existing proposals, since we give formal guarantees that the maximization of $WCC$ delivers cohesive and structured communities. One example of these proposed properties is that communities do not contain bridges. In other words, communities do not include edges that if they were individually removed would disconnect the community. Another interesting property is the linear community cohesion, which states that the number of required connections to include a vertex in a community is directly proportional to the community size. This requirement guarantees that the size of the communities does not affect significantly the density of the community, assuring that both large and small communities are cohesive. Such simple properties, are not plenty fulfilled by the most relevant metrics in the state of the art.

Our third contribution consists of a detectability analysis of $WCC$. We analytically find the detectability threshold of the metric using the *Stochastic Block Model* for graph generation. We show that this threshold fulfills a set of desirable properties, such as that it is independent of the size of the network, that $WCC$ adapts to the inhomogeneities of the graph or that $WCC$ does not detect communities when they do not exist. We empirically validate this detectability threshold and its properties by means of a community detection algorithm based on $WCC$ optimization. Furthermore, we have numerically analyzed and found that $WCC$ is not affected by the so called “community detection paradox”.

As a fourth contribution we propose *Scalable Community Detection (SCD)*, a community detection algorithm based on $WCC$. $SCD$ follows a hill climbing strategy and it is based on a heuristic to estimate how the $WCC$ of a graph partition changes when a vertex is transfered between two communities. The use of this heuristic makes $SCD$ one of the fastest and highest quality community detection algorithm in the state of the art on real graphs.

Finally, we show using real graphs, that there is a correlation between $WCC$ and good communities. We perform a statistical analysis of several indicators, such as the *diameter* and the *edge density*, that are not sufficient to indicate the presence of a community if they are taken alone, but are a good indication of a community structure if they are taken in conjunction. We observe that conductance and modularity are not sufficiently robust and they rank as good communities some which are not.

We experiment with datasets annotated with ground truth communities and several existing community detection algorithms following different strategies and $SCD$. We show that $SCD$ is able to better retrieve those those communities found in the ground truth set while performing fast. We also show that there is a strong correlation between $WCC$ and the annotated data. Finally, we show that $SCD$ is capable of scaling to large real graphs using multiple cores in parallel.

The rest of the paper is structured as follows: in Section 2, we review the state of the art. In Section 3, we introduce the problem of community detection and propose $WCC$. In Section 4, we state the properties of $WCC$ and in Section 5, we show that the current metrics in the state of the art do not fulfill the properties stated. In Section 6 we describe $SCD$. In Section 7, we perform a detectability analysis of $WCC$. In Section 8, we describe the experimental setup. In Section 9, we evaluate the quality of $WCC$ and the performance of $SCD$. Finally, in Section 10 we give guidelines for future work and conclusions.

## 2. RELATED WORK

**Metrics:** There are basically four types of metrics to evaluate the quality of a community. First, those that focus on the internal connectivity of edges in the community. Metrics such as the *average degree*, the *internal edge density*, which is the ratio of the internal number of edges divided by the total possible edges [Radicchi et al. 2004], and the *triangle participation ratio* (TPR), which is the fraction of vertices in the community that closes at least one triangle with two other vertices in the community [Yang and Leskovec 2012], fall into this category. Compared to triangle participation ratio, $WCC$ takes into account not only whether a vertex closes a triangle or not, but also how many triangles are included and their distribution in the community.

The second category of metrics are those that focus on the external connectivity of the community. In this category, we find metrics such as the *cut ratio*, which is the the ratio between the actual number of edges pointing outside the cluster and the total possible number of edges pointing outside the cluster [Fortunato 2010], and the *expansion*, which is the ratio between the number of edges pointing outside the cluster and the size of the cluster [Radicchi et al. 2004].

The third type of metrics, are those that combine both the internal and external connectivity. In this category, we find the *conductance* [Kannan et al. 2004], which is the ratio between the edges going outside the community and the total number of edges between the members of the community, and the *Flake ODF*, which stands for *Flake Out Degree Fraction*, is the average fraction of vertices in the community that have fewer edges pointing inside than outside of the community [Flake et al. 2000].

Finally, the fourth category of metrics are those that measure the quality of a community compared to a network model. The most popular metric that falls into this category is the *modularity*, which was proposed in [Newman and Girvan 2004] and is probably the most widely used metric in the state of the art. Modularity measures the internal connectivity of the community compared to that expected in random graph with the same exact degree sequence. Modularity has become very popular in the literature, and a lot of algorithms are based on its maximization. However, it has been reported that modularity has resolution limits [Fortunato and Barthélemy 2007; Good et al. 2010]. Communities detected by modularity depend on the total graph size, and thus, for large graphs, small well defined communities are never found. This means that maximizing the modularity leads to partitions where communities are far from intuitive.

A recent survey [Leskovec et al. 2010] of community metrics discusses the performance of metrics on real networks. In this survey, Leskovec et al. showed that, among all these metrics, conductance is the metric that best captures the concept of community (modularity is not included in the analysis). In [Yang and Leskovec 2012], the authors use ground truth data to determine that both the conductance and the TPR are those metrics that best capture the concept of real world communities.

Broadly speaking, existing metrics focus on maximizing or minimizing certain aggregated values or ratios, without paying attention to the internal/external structure of the communities. Indeed, the only metric that takes a more structured approach, such as TPR, has proven to be one of the most robust metrics.

**Algorithms:** The most common category of algorithms is formed by those based on maximizing modularity. In the literature, we find several proposals based on different optimization strategies such as agglomerative greedy [Clauset et al. 2004], simulated annealing [Medus et al. 2005] or multistep [Blondel et al. 2008] approaches, just to cite a few of them.

Regarding the algorithms not based on maximizing modularity, we find Infomap, based on performing random walks [Rosvall and Bergstrom 2008], an algorithm based on computing the edge clustering coefficient [Radicchi et al. 2004], the clique percolation method, based on computing chains of cliques [Palla et al. 2005], or the Label Propagation Method [Raghavan and Albert 2007], which has acquired a great popularity due to its linear complexity. According to [Lancichinetti 2009], the best community detection algorithm in the state of the art is Infomap.

Finally, there is a category of algorithms, such as OCA [Padrol-Sureda et al. 2010], Link Clustering [Ahn et al. 2010], Oslom [Lancichinetti et al. 2011] and BigClam [Yang and Leskovec 2013], that aim at looking at overlapping communities, that is, communities that can share vertices with other communities or core-periphery communities [Yang and Leskovec 2014], where these are formed by a core and a periphery. However, in this paper we focus on the search of disjoint communities, where each vertex can only belong to one community.

## 3. WEIGHTED COMMUNITY CLUSTERING

### 3.1. Problem Formalization

Given a graph $G = (V, E)$, the problem of disjoint community detection consists in classifying the $|V| = n$ vertices of the graph into $q$[1] non-empty pairwise disjoint cohesive sets, $S_i$ for $1 \leq i \leq q$. We call those $q$ sets a partition of $V$, i.e. $\mathcal{P} = \{S_1, \dots, S_q\}$, in such a way that $S_1 \cup \dots \cup S_q = V$.

The criterion to measure the degree of cohesion of each set is formally obtained by defining a metric, that is, a function $f_s$, that assigns a real number to each subset $S_i$ of $V$ such that $0 \leq f_s(S_i) \leq 1$. A good/bad *community* is a set of vertices $S$ with a value of $f_s$ close to 1/0.

We define the cohesion of a community $f_s(S)$ as the average cohesion of its vertex members $x$ with respect to the set $S$:

$$f_s(S) = \frac{1}{|S|} \sum_{x \in S} f_v(x, S). \tag{1}$$

Similarly, we define the metric on a partition $\mathcal{P}$ by taking the weighted average of the value of the function on the sets $S_i$ of the partition:

$$f(\mathcal{P}) = \frac{1}{n} \sum_{i=1}^{q} \left( |S_i| \cdot f_s(S_i) \right). \tag{2}$$

For a given graph and a given metric $f$ in $G$, the goal is to obtain an *optimal partition*, that is, a partition $\mathcal{P}$ such that $f(\mathcal{P})$ takes a maximum value. We call the communities in an optimal partition the *optimal communities* of the graph.

### 3.2. Metric Definition

The difficulty of community detection arises from accurately defining $f(S)$, that is, how the cohesion of a set $S$ is quantified. According to the informal community definition, cohesive sets are those that encompass the two following features:

- The set is structurally isolated from the rest of the graph (i.e the external connectivity of the set is small).
- The set is structurally intraconnected (i.e. the internal connectivity of the set is large).

Therefore, it is reasonable to design an $f_s(S)$ to encompass these two characteristics. If $f_s(S)$ is too biased towards promoting the structural isolation of the set (i.e. it scores those sets with a small external connectivity high, but ignores the internal connectivity), then vertices which are barely connected to the rest of the graph might be included in the

[1] We assume that $q$ is a value determined by the nature of the graph rather than an arbitrary value determined by the user

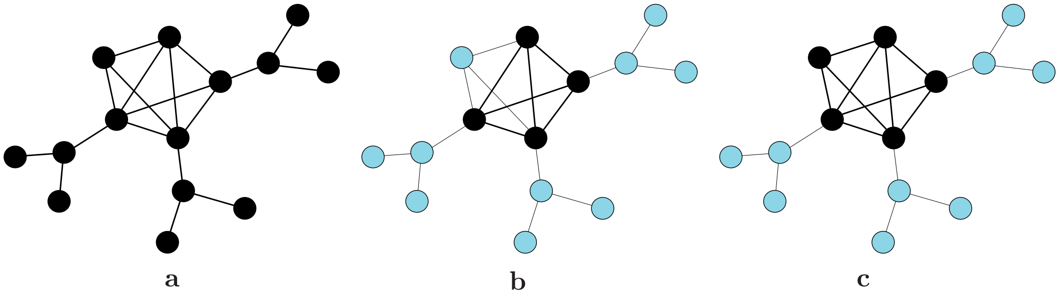


Fig. 1. **a)** Metric maximizing structural isolation. **b)** Metric maximizing structural intraconnectivity. **c)** Metric taking a compromise between structural isolation and intraconnectivity

same community. This is exemplified in Figure 1(a), where considering the whole graph as a community would obtain the best score with such a metric. On the other hand, if $f_s(S)$ is biased towards promoting the structural intraconnectivity of the set (i.e. scoring high sets with an extreme internal connectivity, but ignoring the external connectivity), then some considerably affine vertices with those in $S$ might not be included in community $S$. Taking this statement to the limit, we would only select the maximal cliques in the graph. This is exemplified in Figure 1(b), where the set of four vertices in black would have a better score than the whole dense region in the middle of the graph. A good balance between these two characteristics is crucial to properly identifying the communities as exemplified in Figure 1(c).

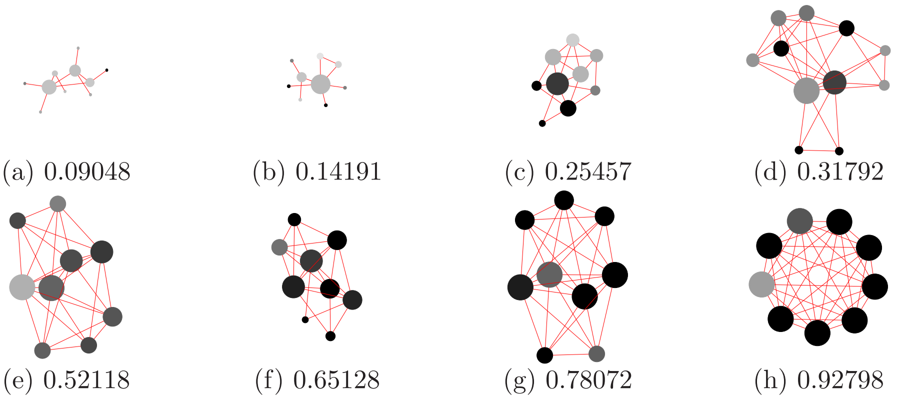
Last but not least, the previously stated features rely on the concept of structure. Traditionally, the presence of a single edge between two vertices has been typically taken as a sufficient condition to consider that two vertices are structurally connected, but this is not sufficient for us.

Thus, in this paper, we take a different definition for structure that relies on the "homophily principle" that is latent in social networks [McPherson et al. 2001]. In short, this principle says that similar entities tend to establish connections among them. In social networks, people are more likely to connect to other people that work at the same place, have similar interests or have strong social interactions. The consequence is that the people that form these communities are very connected and inside these communities many triples of vertices are connected by triangles. In this paper, we suggest to use triangles to identify the structured connections of a community, and hence differentiate them from casual connections.

With this in mind, we design a community metric sensitive to both the structural isolation and the intraconnectivity, and which takes the triangle as the basic indicator of structure. We denote by $t(x, S)$ the number of triangles that vertex $x$ closes with the vertices in a set $S$ and by $vt(x, S)$ the number of vertices of $S$ that form at least one triangle with $x$. We propose the Weighted Clustering Coefficient, $WCC_v(x, S)$, as the specific implementation of $f_v(x, S)$ in Equation 1:

$$WCC_v(x,S) = \begin{cases} 0 & \text{if } t(x,V) = 0 \\ \underbrace{\frac{t(x,S)}{t(x,V)}}_{isolation} \cdot \underbrace{\frac{vt(x,V)}{vt(x,V) + |S \setminus \{x\}| - vt(x,S)}}_{intraconnectivity} & \text{if } t(x,V) \neq 0. \end{cases} \quad (3)$$

Note that $vt(x, V) + |S \setminus \{x\}| - vt(x, S) = 0$ implies that $S = \{x\}$ and $vt(x, V) = 0$. Then, condition $vt(x, V) + |S \setminus \{x\}| - vt(x, S) = 0$ is included in condition $t(x, V) = 0$.



(a) 0.09048 (b) 0.14191 (c) 0.25457 (d) 0.31792

(e) 0.52118 (f) 0.65128 (g) 0.78072 (h) 0.92798

Fig. 2. Examples of communities from real graphs, sorted by $WCC_s$.

The left factor of $WCC_v(x, S)$ is the ratio of triangles that vertex $x$ closes with set $S$, as opposed to the number of triangles that $x$ closes with the whole graph. The left factor is maximized for a vertex $x$ when $S$ includes *all* the vertices that form triangles with $x$. Note that since a pair of vertices can build many triangles, the left term rewards the inclusion of the vertices that build more triangles with $x$. In other words, it measures the structural isolation of vertex $x$.

On the other hand, the right factor is the ratio between the number of vertices in $V$ that close at least one triangle with $x$, and the number of vertices in $V$ that close at least one triangle with $x$ plus the number of those in $S \setminus \{x\}$ not closing any triangle with $x$ and another vertex $u \in S \setminus \{x\}$. The right term is maximized for $x$ when $S$ contains *only* vertices that do form at least one triangle with $x$ and a third vertex $u \in S$. In other words, it measures the structural intraconnection of vertex $x$ with set $S$.

The two factors of $WCC_v(x, S)$ are finally combined with a multiplication because it is necessary to maximize both concepts. If any of the two terms is zero the cohesion of the vertex with respect to the set is zero.

Finally, analogously to $f_s(S)$ in Equation 1, we denote the quality of a community as $WCC_s(S)$. Figure 2 shows some examples of communities with different values of $WCC_s$, showing different levels of cohesion. These communities are extracted randomly from the set of communities found in the real graphs by the algorithms in Section 9. The color of the vertices represents the percentage of neighbors belonging to the community. The darker the vertex, the larger the percentage of neighbors of the vertex that belong to the community, that is, the larger the *isolation* of the vertex. On the other hand, the size of the vertices represents the percentage of vertices of the community that are actual neighbors of that vertex. The larger the size of the vertex, the more connected the vertex is with the other vertices of the community, that is, the larger the *intraconnectivity*. In other words, the color represents the isolation, while the size represents the intraconnections. Thus, the better the community is, the larger and darker are its vertices. We see then, that there is a correlation between high $WCC_s$ values and good communities.

### 3.3. Basic behavior

We formally summarize the basic behavior of $WCC_v(x, S)^2$:

PROPOSITION 3.1. *Let $G = (V, E)$ be a graph and $\emptyset \neq S \subseteq V$. Then,*

(i) $0 \leq WCC_v(x, S) \leq 1$ *for all* $x \in V$.

[2] All the proofs of the propositions and theorems introduced in this paper can be found in the Appendix.

(ii) $WCC_v(x, S) = 0$ *if and only if* $t(x, S) = 0$.
(iii) $WCC_v(x, S) = 1$ *if and only if* $vt(x, V) = vt(x, S) = |S \setminus \{x\}| \geq 2$.

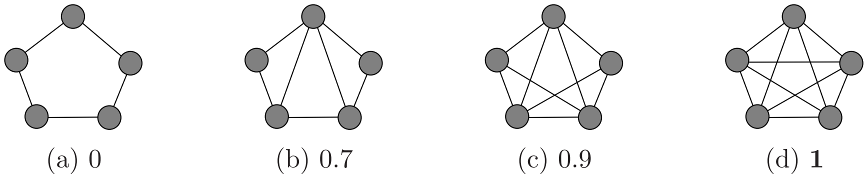
The value of $WCC_v(x, S)$ indicates the cohesion of vertex $x$ with respect to $S$. This value is a real number between 0 and 1 (Proposition 3.1 (i)). These two extreme values are only observed in particular situations (Proposition 3.1 (ii-iii)). On the one hand, for a given vertex $x$, in order to have some degree of cohesion with a subset $S$, the vertex must at least form one triangle with two other vertices in set $S$. If a vertex builds no triangle with the vertices in $S$, then the cohesion of the vertex with respect to the set is zero. On the other hand, value one is reached if and only if $S$ includes exactly and only all the vertices that close triangles with $x$. Furthermore, from the point of view of $WCC_v$, only those edges in $E$ closing at least one triangle are relevant and influence the cohesion of a vertex.

We infer three characteristics on $WCC_s(S)$ from Proposition 3.1 as follows.

PROPOSITION 3.2. *Let* $G = (V, E)$ *be a graph and* $\emptyset \neq S \subseteq V$. *Then,*

(i) $0 \leq WCC_s(S) \leq 1$.
(ii) $WCC_s(S) = 0$ *if and only if* $S$ *has no triangles.*
(iii) $WCC_s(S) = 1$ *if and only if* $S$ *is a clique with* $vt(x, V) = vt(x, S)$ *for all* $x \in S$.

The clique is the subgraph structure that best resembles the perfect community, and thus, $WCC_s$ rates it with the largest value. On the other hand, if the community has no triangles, its quality is the minimum possible. In Figure 3(a-d), we show a community of five vertices with an increasing number of internal triangles. The larger the density of triangles, the larger the $WCC_s$ value for the community.



(a) 0 (b) 0.7 (c) 0.9 (d) **1**

Fig. 3. Example of the sensitivity of $WCC$ against triangles

Finally, according to Equations 1 and 2, an optimal partition is such that, for all vertices of the graph, function $WCC_v(x, S)$ is optimized.

## 4. PROPERTIES OF $WCC$

Following the previous requirements of structural isolation and intraconnectivity, many formulas may be composed by combining different measures of such requirements in Equation 3. However, this is not a sufficient condition to ensure a good community metric. In this section, we present a set of properties that $WCC$ meets, giving additional insight of the behavior and robustness of the metric. We present these as generic properties that we think are worth to be considered when designing any good community detection metric.

### 4.1. Property 1: Internal Structure

Typically, existing community metrics take all the internal edges of a community considering they are equally important, without considering whether they form structures or not. The structure of the edges connecting the vertices is important to determine whether these form a good community or not. Therefore, ***the cohesion of a given community ought to depend, not only on the number of internal edges, but on how those edges connect the vertices of the community.***

One such structure is the triangle, as stated above, which indicates the presence of a strong relation in the graph. The relevance of triangles in social networks has been confirmed in previous studies [Newman and Park 2003; Newman 2001; Shi et al. 2007; Satuluri et al. 2011] and models describing the growth of social networks give triangle closing as a key factor of network evolution [Leskovec et al. 2008]. $WCC$ is crafted to be sensitive to triangles, and as a consequence, to be sensitive to the internal structure of the community. We verify this property for $WCC$: the left factor in Equation (3) is the ratio of the number of triangles that vertex $x$ forms with the vertices in $S$ as opposed to the number of triangles that vertex $x$ forms with the whole graph. Hence, this left factor is affected by the number of triangles inside the community. On the other hand, the right factor depends on the number of vertices that form triangles with vertex $x$. Therefore, the distribution of triangles inside the community affects the right factor.

Figure 4 shows two examples of communities with the same number of vertices and edges, but distributed differently. While in Figure 4(a) we see two cliques with only three edges connecting them, in Figure 4(b) we see a more uniformly structured community closing more triangles. We see that $WCC_s$ scores Figure 4(b) higher, since the community is more structurally intraconnected, eventhough there are two cliques in Figure 4(a).

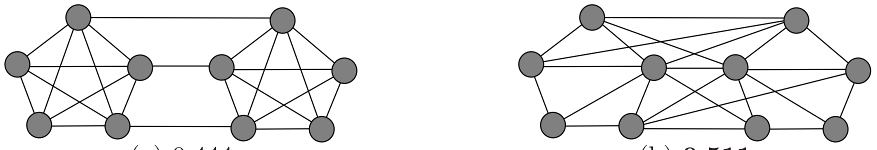

(a) 0.444 (b) **0.511**

Fig. 4. Consequences of the internal structure on the $WCC$

### 4.2. Property 2: Linear Community Cohesion

Communities are groups of vertices with a significant level of cohesion, that is, the number of triangles closed by the vertices forming the community is high. This means that, as long as the size of the community increases, the number of links between a vertex and a community has to increase in order to maintain the level of cohesion of the community. This simple restriction limits the community growth if there is not a significant cohesion among its members. Therefore, ***the number of connections needed between a vertex $x$ and a set $S$, so that $f(\{S \cup \{x\}\}) \geq f(\{S, \{x\}\})$, must grow linearly with respect to the size of $S$***. If it grew sublinearly, the larger the communities, the easier it would be for a vertex to join that community with respect to the size the community. On the other hand, if it grew faster than linear, the communities would have a maximum possible size, since after a certain point, the number of necessary links between a vertex and the rest of the community would be larger than the possible number of links.

$WCC$ is crafted to verify this property by means of he following theorem:

THEOREM 1. *Let $G = (V, E)$ be a random graph of order $r$ in which each edge occurs independently with probability $p$ and closes at least one triangle. Let $v \notin V$ be a vertex connected to and forming at least one triangle with $d \geq 2$ vertices of $V$. Consider the two partitions $\mathcal{P}_1 = \{V \cup \{v\}\}$ and $\mathcal{P}_2 = \{V, \{v\}\}$. Then,*

(i) $(r+1)WCC(\mathcal{P}_1) = (r-1)p + 2dr^{-1}$.

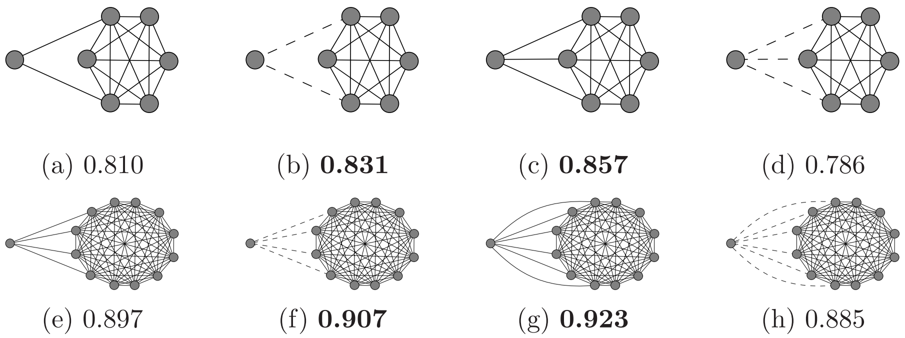
(ii) $(r+1)WCC(\mathcal{P}_2) = (r-d)p + \dfrac{d}{r} \cdot \dfrac{((r-1)p+1)(r-1)(r-2)p^2}{(r-1)(r-2)p^2 + 2(d-1)}$.

(iii) *For $r$ large enough, $WCC(\mathcal{P}_1) > WCC(\mathcal{P}_2)$ if and only if*
$d > rp\left(\sqrt{p^2 + 2p + 9} - (1 + p)\right)/4.$

For instance, in the particular case of the clique (where $p = 1$), it is necessary to connect to roughly more than one third of the vertices to become a member of the community.

COROLLARY 1. *Let $S$ be a clique of order $r$. Given a vertex $v$, there must exist at least $0.37 \cdot r$ edges between $v$ and $S$ to hold $WCC(\{S \cup \{v\}\}) > WCC(\{S, \{v\}\})$.*

In Figure 5 we show an example of Theorem 1, where we represent four groups of examples: a-b, c-d, e-g and g-h. The left graph of each group (a,c,e and g), represents a partition with one single community, while the right graph assumes a partition with two distinct communities formed by a single vertex and a clique. We see that group a-b has a better $WCC$ score for distinct communities while group c-d for one community. Note that $WCC$ gives a better score for vertices connected to more than 0.37 vertices of a clique. The same happens in examples e-f (four connections are less than 0.37 vertices), and g-h (6 connections are more than 0.37 vertices). This example illustrates the linear community cohesion of $WCC$, where the number of connections required by a vertex to become part of a community, scales with its size.


(a) 0.810 (b) **0.831** (c) **0.857** (d) 0.786

(e) 0.897 (f) **0.907** (g) **0.923** (h) 0.885

Fig. 5. Examples showing the linear community cohesion of $WCC$

We empirically validated the lineality property of $WCC$. We generated instances of graphs formed by a vertex $v$ and a community $C$, with a probability $p_{in}$ for an edge to exist between two vertices of $C$, and a probability $p_{out}$ for an edge to exist between $v$ and a vertex of $C$. We generated two types of graphs: one with a community of size 100, and another one with a community of size 200. For each possible configuraion of $p_{in}$ and $p_{out}$ (in steps of 0.01), we generated 100 instances of each type. Using the $WCC$ based algorithm proposed in Section 6, we tested for which configurations the resulting partition was formed by a single community containing both the original community and the vertex, and which did not. Figure 6(a) and (b) show the theoretical threshold line of Theorem 1 for both sizes in dim white, and in grey scale the results obtained by the algorithm. White means that the algorithm opted to merge the communities into a single one for all the instances of that particular configuration, while black means that a non merging configuration was always returned. From the figure we can observe that the theoretical threshold is empirically observed, becoming sharper as the size of the graph increases. The black region at the bottom of Figure 6 becomes smaller as the graph grows, and as predicted in Theorem 1, it disapears for arbitrarily large graphs. For small graphs and small $p_{in}$, the probability for a vertex to exist without closing any triangle is large (especially the smaller $p_{out}$ is), and therefore the community is shattered into smaller sub communities.

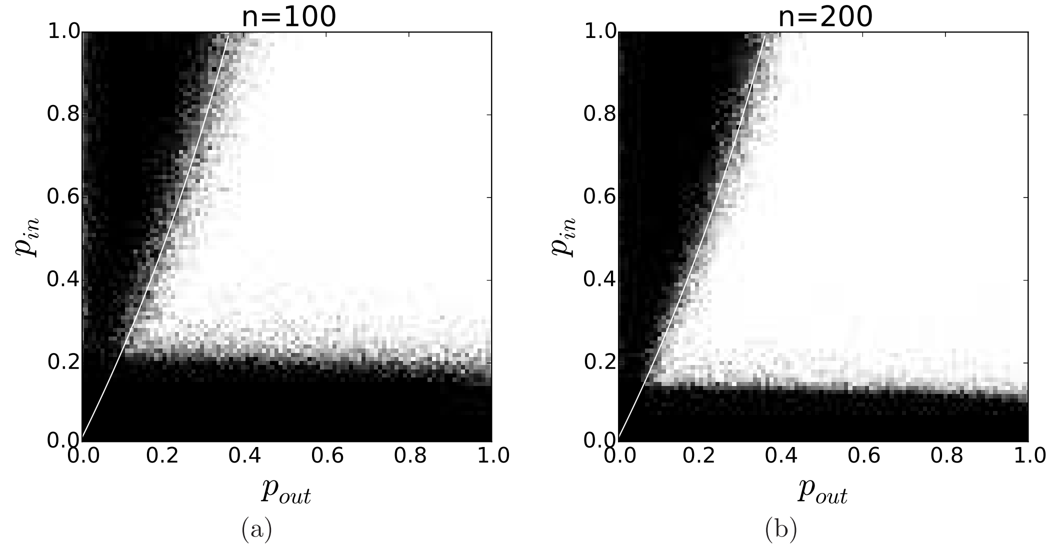


Fig. 6. Empirical evaluation of Property 2.

### 4.3. Property 3: Bridges

The connections in real graphs are known not to be local, but can connect distant vertices [Liben-Nowell and Kleinberg 2007]. A bridge is an edge that if it is removed from the graph, it creates two connected separate components. A bridge is a very weak relation between two sets of vertices that are unrelated, because it only affects one member of both subsets of vertices. Therefore, ***an optimal community in social networks can not contain a bridge.*** Optimal communities found by *WCC never* contain bridges. We show this based on the following observation:

THEOREM 2. *Let $S_1$ and $S_2$ be two communities in a partition of graph $G = (V, E)$ such that:*

(i) *$S_1$ and $S_2$ are the set of vertices of two different connected components.*
(ii) *$WCC_s(S_1) > 0$.*

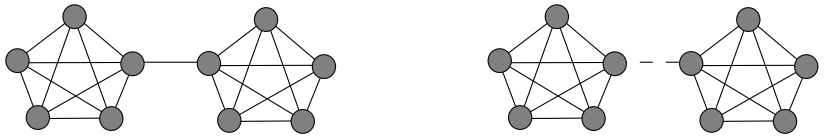
*Then, the following inequality holds:*

$$WCC(\{S_1, S_2\}) > WCC(\{S_1 \cup S_2\}).$$

When an edge does not close any triangle, it does not affect the computation of $WCC$. A bridge does not close any triangle, hence, a bridge is never accounted by $WCC$. Thus, sets of vertices connected by bridges are not merged into a community because of Theorem 2. In Figure 7 (a-b), we show an example of the application of Theorem 2. We see that having the two cliques separated is better than considering a single community with a bridge, in terms of $WCC$.

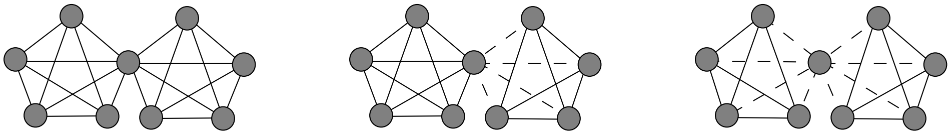
### 4.4. Property 4: Cut Vertex.

A vertex is a cut vertex when its removal separates the graph into two (or more) connected components. Cut vertices are weak links in a community, because there are no edges between the separated components, and thus the components have no structural connectivity. If the separated components have a strong internal structure, then it is more natural to split the



(a) 0.444 (b) **1**

Fig. 7. Example of the behavior of $WCC$ against bridges



(a) 0.556 (b) **0.722** (c) 0.444

Fig. 8. Example of $WCC$ against vertex cuts

components into several communities. Therefore, we require that ***an optimal community does not contain a cut vertex that separates two structurally intraconnected sets***[3]. In Figure 8(a-c), we show two cliques (note that the clique is the highest density graph structure) of size five sharing a vertex. Here, $WCC$ is able to separate the communities for this particular case because the left and right sets of vertices have enough structural intraconnectivity and isolation to become separate communities, assigning the cut vertex to one of them. We prove this property for $WCC$ for the case where communities have the highest possible density, which is the clique:

THEOREM 3. *Let $G = (V, E)$ be a graph of order $n$ which consists of two cliques $K_r$ and $K_s$ of orders $r$ and $s$, respectively, that intersect in a vertex $t$. Assume $r \geq s \geq 4$.*

(i) *If $\mathcal{P}_1 = \{K_r \cup K_s\}$, then*

$$n \cdot WCC(\mathcal{P}_1) = \frac{(r-1)(r-1)}{r+s-2} + \frac{1}{r+s-2} + \frac{(s-1)(s-1)}{r+s-2}; \tag{4}$$

(ii) *if $\mathcal{P}_2 = \{K_r,\ K_s \setminus \{t\}\}$, then*

$$n \cdot WCC(\mathcal{P}_2) = (r-1) + \frac{(r-1)(r-2)}{(r-1)(r-2)+(s-1)(s-2)} + \frac{(s-1)(s-2)(s-3)}{(s-1)(s-2)}; \tag{5}$$

(iii) *if $\mathcal{P}_3 = \{K_r \setminus \{t\}, \{t\},\ K_s \setminus \{t\}$, then*

$$n \cdot WCC(\mathcal{P}_3) = \frac{(r-1)(r-2)(r-3)}{(r-1)(r-2)} + \frac{(s-1)(s-2)(s-3)}{(s-1)(s-2)}; \tag{6}$$

(iv) $WCC(\mathcal{P}_3) \leq WCC(\mathcal{P}_2)$.

(v) $\max\{WCC(\mathcal{P}_1), WCC(\mathcal{P}_2), WCC(\mathcal{P}_3)\} = WCC(\mathcal{P}_2)$.

This theorem illustrates the fact that $WCC$ avoids merging two very well defined communities (such as two cliques) because of a single vertex. The reason is that $WCC$ is a metric that not only takes into account the vertices that are connected and forms triangles, but also the vertices that do not. Thus, if the triangles inside the community are not distributed evenly among all the vertices then the quality of the community is penalized.

[3] A vertex cut can be seen as an example of an overlapped community. However, it is not the aim of this work to consider the problem of overlapping communities.

## 5. COMPARISON WITH OTHER METRICS

In the state of the art we find other metrics which attempt to quantify the quality of a community. However, these metrics do not fulfill one or more of the properties proposed in this paper, which are fulfilled by $WCC$. Therefore, optimizing those metrics locally or globally does not guarantee structured communities.

**Cut ratio and expansion**: They are based on the external connectivity of the community, that is, the lower the number of connections pointing outside of the community, the better. Since, they do not pay attention to the internal connectivity, these metrics do not fulfill Property 1, which means that they do not care about how the vertices in the community are connected. Furthermore, cut ratio and expansion have an optimization problem. Given a graph, the partition consisting of a single community containing all the vertices of the graph obtains always the optimal score. This optimization problem implies that that properties 2, 3 and 4 are not fulfilled.

**Conductance and Flake ODF**: They are based on the internal versus the external connectivity of the community. Although these metrics consider the internal edge density, they do not consider how the internal edges are actually distributed, and hence they do not fulfill Property 1. Conductance and Flake ODF have the same optimization problem as cut ratio and expansion and thus, they neither fulfill properties 2, 3 and 4.

**Internal edge density and TPR**: Internal edge density and TPR focus on the internal connectivity of the community. Internal edge density, like conductance and Flake ODF, does not fulfill Property 1 because it does not consider the internal distribution of the edges. On the other hand, internal edge density fulfills properties 2, 3 and 4. However, optimizing internal edge density becomes problematic because it would only find maximal cliques, which is too restrictive. Regarding TPR, it fulfills property 1, since it is dependant on the triangles closed in the community and property 3, because closing one triangle with the community implies closing a cycle of size three. However, it does not fulfill property 2, because only a triangle between a vertex and a community is required for that vertex to become part of the community. Finally property 4 is not fulfilled because as long as the cut vertex forms a triangle with each of the communities it connects, the communities can be taken as a single community according to the metric. Although TPR is similar to $WCC$ in the sense that it is based on triangles, it lacks the precision and robustness that $WCC$ has, as we will see in Section 9.1.

**Modularity**: Modularity suffers from resolution limits [Fortunato and Barthélemy 2007; Good et al. 2010]. This resolution problem is exemplified in Figure 9, where the optimal communities for modularity are groups of two cliques. In this example, the communities with the optimal modularity contain a bridge, and thus they do not verify Property 3. However, the natural communities which are the groups of five vertices forming cliques are the optimal communities for $WCC$. The $WCC$ value of the five vertices clique is one, so having a partition with each clique as a community has the maximum $WCC$ value. Furthermore, is has been shown [Bagrow 2012] that trees, can have arbitrarily large modularity, while $WCC$ scores them with zero since they cannot be intuitively considered communities. We show that $WCC$ is a metric that sees the communities in a local fashion, focusing on the internal density and the connections with their surroundings instead of the whole graph. Modularity assumes that graphs are homogeneous, whereas they are not.

## 6. COMMUNITY DETECTION ALGORITHM

In this section we describe Scalable Community Detection (SCD), a community detection algorithm based on $WCC$ and designed to scale on SMP machines. SCD takes a graph $G$ as input, and generates a partition of $G$ resulting from a $WCC$ optimization process.

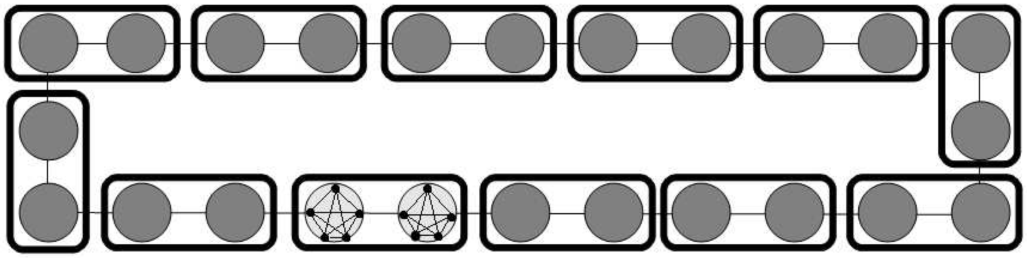


Fig. 9. Ring with 24 cliques of 5 vertices each (shaded circles). Setting each clique as a community has a modularity of 0.8674, but merging adjacent cliques has modularity 0.8712 [Good et al. 2010].

The algorithm is divided into three phases: graph cleanup, initial partition and partition refinement, which are described in the following paragraphs.

### 6.1. Graph cleanup

Once the graph is loaded into memory, we perform a cleanup process aimed at removing the unnecessary edges and computing a set of statistics that will be helpful during the next phases. The process consists of computing first the number of triangles each edge in the graph closes. Then, we remove from the graph those edges that do not close any triangle, as these are irrelevant from the point of view of $WCC$, that is, do not have any effect during the $WCC$ computation. By removing these edges we reduce the memory consumption and improve the performance of SCD. Furthermore, we can also simplify the heuristic proposed in Section 6.4 (which we use to improve the performance of the partition refinement step) since we can assume that each edge closes at least one triangle. Among the statistics computed are the clustering coefficient of the graph and each vertex, and the number of triangles each vertex belongs to.

### 6.2. Initial partition

The goal of this step is to find an initial partition which can be later refined. This initial partition is computed by a fast heuristic process, described in Algorithm 1. We first sort the vertices of the graph by their clustering coefficient decreasingly (which was computed during the graph cleanup step). For those vertices with equal clustering coefficient, we use the degree as a second sorting criterion (Line 2). Then, the vertices are iterated and, for each vertex $v$ not previously visited, we create a new community $C$ that contains $v$ and all the neighbors of $v$ that were not visited previously (Line 6 to 12). Finally, community $C$ is added to partition $P$ (Line 13) and all the vertices of the community are marked as visited. The process finishes when all the vertices in the graph have been visited.

This heuristic is built on top of the following intuition: the larger the clustering coefficient of a vertex, the larger the number of triangles the vertex closes with its neighbors, and the larger the probability that its neighbors form triangles among them. Hence, considering Equation 3, the larger the clustering coefficient of a vertex, the larger is the probability that the $WCC$ of its neighbors is large if we include them in the same community.

### 6.3. Partition refinement

Algorithm 2 describes the partition refinement step. It takes the initial partition computed in the "Initial partition" step and refines it by following a hill climbing strategy, that is, in each iteration, a new partition is computed from the previous one by performing a set of modifications (movements of vertices between communities) aimed at improving the $WCC$ of the new partition. The algorithm repeats the process until the $WCC$ of the new partition does not percentually improve over the best $WCC$ observed so far more than

**ALGORITHM 1:** Phase 1, initial partition.

```
Data: Given a graph G(V,E)
Result: Computes a partition of G
Let $P$ be a set of sets of vertices;
S ← sortByCC( V );
foreach v in S do
    if not visited(v) then
        markAsVisited(v);
        C ← v;
        foreach u in neighbors(v) do
            if not visited(u) then
                markAsVisited(u);
                C.add(u);
            end
        end
        S.add(C);
    end
end
return P;
```

a given threshold, and a set of *lookahead* iterations has been performed. These *lookahead* iterations are used to make the algorithm more robust against local maxima. In our tests, setting the threshold to 1% and the *lookahead* to five iterations provided a good tradeoff between performance and quality.

In each iteration, for each vertex $v$ of the graph, we use the `bestMovement` function to compute the movement of $v$ that improves most the $WCC$ of the partition most (Line 8). There are four types of possible movements:

— **NO_ACTION**: leave the vertex in the community where it currently is.
— **INSERT**: insert a singleton vertex into an existing community. Remove the empty community resulting from this movement.
— **REMOVE**: remove the vertex from its current community and create a new singleton community formed by the vertex.
— **TRANSFER**: remove the vertex from its current community (source) and insert it into another one (destination).

Note that `bestMovement` does not modify the current partition, and that the best movement of each vertex is computed independently from the others. This allows computing in parallel the best movements for all the vertices. Once we compute the best movement of all the vertices of the graph, we apply all of them *simultaneously* (`applyMovements` Line 10). Finally, we update the $WCC$ of the new partition (Line 11) and check whether it improved compared to the last iteration.

Before describing function `bestMovement` in detail, we first introduce some auxiliary functions that are used in it. The proofs of the theorems introduced in this section can be found in the Appendix.

— $WCC_I(v, C, P)$ computes the improvement of the $WCC$ of a partition $P$ when vertex $v$ (which belongs to a singleton community of $P$) is inserted into community C of $P$.

**ALGORITHM 2:** Phase 2, refinement.

```
Data: Given a graph G(V,E) and a partition P
Result: A refined partition P'
bestP ← P;
bestWCC ← computeWCC(P);
triesRemaining ← lookAhead;
repeat
    triesRemaining - -;
    M ← ∅;
    foreach  $v$ $in$ $V$ do
        M.add( bestMovement(v,P) );
    end
    P ← applyMovements(M,P);
    newWCC ← computeWCC(P);
    if  $(newWCC - bestWCC)/bestWCC \geq t$ then
        bestP ← P;
        bestWCC ← newWCC;
        triesRemaining ← lookAhead;
    end
until  $triesRemaining > 0$;
return bestP;
```

THEOREM 4. *Let $P = \{C_1, C_2, \ldots, C_k, \{v\}\}$ and $P' = \{C'_1, C_2, \ldots, C_k\}$ be partitions of a graph $G = (V, E)$ where $C'_1 = C_1 \cup \{v\}$. Then,*

$$WCC(P') - WCC(P) = WCC_I(v, C_1, P) =$$
$$= \frac{1}{|V|} \cdot \sum\nolimits_{x \in C_1} [WCC(x, C'_1) - WCC(x, C_1)] + \frac{1}{|V|} \cdot WCC(v, C'_1).$$

- $WCC_R(v, C, P)$ computes the improvement of the $WCC$ of a partition $P$ when vertex $v$ is removed from community $C$ of $P$ and placed as a singleton community.

THEOREM 5. *Let partitions $P = \{C_1, C_2, \ldots, C_k\}$ and $P' = \{C'_1, C_2, \ldots, C_k, \{v\}\}$ of a graph $G = (V, E)$ where $C_1 = C'_1 \cup \{v\}$. Then,*

$$WCC(P') - WCC(P) = WCC_R(v, C_1, P) = -WCC_I(v, C'_1, P').$$

- $WCC_T(v, C_1, C_2, P)$ computes the improvement of the $WCC$ of a partition when vertex $v$ is transfered from community $C_1$ and to $C_2$.

THEOREM 6. *Let $P = \{C_1, C_2, \ldots, C_{k-1}, C_k\}$, $P' = \{C'_1, C_2, \ldots, C_{k-1}, C_k, \{v\}\}$ and $P'' = \{C'_1, C_2, \ldots, C_{k-1}, C'_k\}$ be partitions of a graph $G = (V, E)$ where $C_1 = C'_1 \cup \{v\}$ and $C'_k = C_k \cup \{v\}$. Then,*

$$\begin{aligned} WCC(P'') - WCC(P) &= WCC_T(v, C_1, C_k, P) \\ &= WCC_R(v, C_1, P) + WCC_I(v, C_k, P). \\ &= -WCC_I(v, C'_1, P') + WCC_I(v, C_k, P). \end{aligned}$$

**ALGORITHM 3:** `bestMovement`.

```
   Data: Given a graph G(V,E) a partition P and a vertex v
   Result: Computes the best movement of v.
 1 m ← [NO_ACTION];
 2 sourceC ← GetCommunity(v,P);
 3 $wcc\_r$ ← $WCC_R$(v,sourceC,P);
 4 $wcc\_t$ ← 0.0;
 5 bestC ← ∅;
 6 Candidates ← candidateCommunities(v,P);
 7 for c in Candidates do
 8     if size(sourceC) > 1 then
 9         aux ← $WCC_T$(v,sourceC,c,P) ;
10     else
11         aux ← $WCC_I$(v,c,P) ;
12     end
13     if aux > $wcc\_t$ then
14         $wcc\_t$ ← aux;
15         bestC ← c;
16     end
17 end
18 if $wcc\_r$ > $wcc\_t$ and $wcc\_r$ > 0.0 then
19     m ← [REMOVE];
20 else if $wcc\_t$ > 0.0 then
21     if size(sourceC) > 1 then
22         m ← [TRANSFER , bestC];
23     else
24         m ← [INSERT , bestC];
25     end
26 end
27 return m;
```

From Theorem 4, we conclude that computing the improvement of $WCC$ resulting from inserting a vertex $v$ (i.e. a singleton community) into a community $C$, we only need to recompute the $WCC$ of vertex $v$ and those vertices in $C$. Therefore, when computing $WCC_I()$ for a vertex and a community, only a very local portion of the graph needs to be accessed, and the number of computations performed is small compared to computing the $WCC$ of the whole partition. Furthermore, Theorems 5 and 6 show that we can express any of the movements needed by the algorithm (more concretelly INSERT, REMOVE and TRANSFER), in terms of function $WCC_I()$, which in turn simplifies the implementation of the algorithm.

Algorithm 3 describes the `bestMovement` function. First, we compute the improvement of removing vertex $v$ from its current community (Line 3). Then, we obtain the set of candidate communities, formed by those communities containing the neighbors of $v$ (Line 6). After that, we calculate which is the candidate community where inserting or transferring vertex $v$ (depending whether the $v$ forms a singleton community or not) improves the $WCC$ most (Lines 7 to 17). Finally, we select whether the best improvement is obtained from removing the vertex from its current community (REMOVE) or inserting/transferring it into a new community (INSERT/TRANSFER) (Lines 18 to 26). If neither of the two movements improves the $WCC$ of the partition, we keep the vertex in the current community (NO_ACTION) (Line 1).

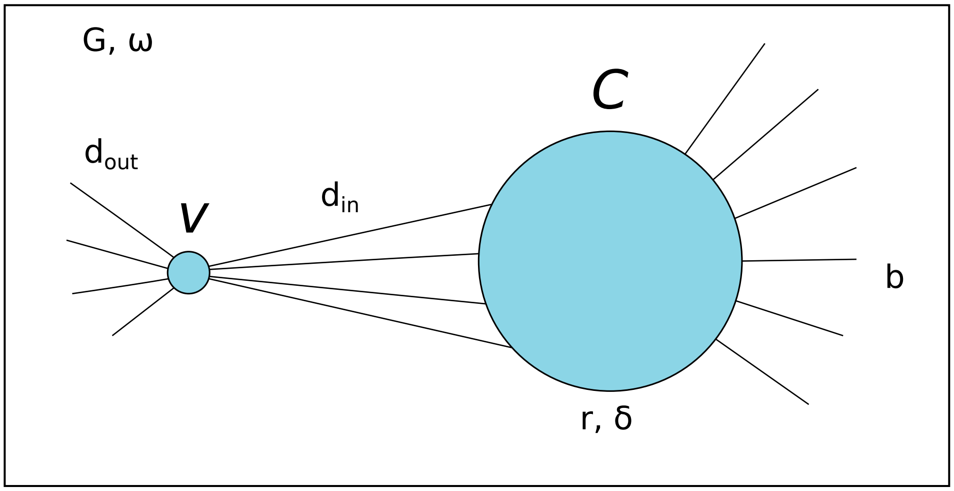


Fig. 10. Model used for estimating the $WCC_I$.

### 6.4. WCC$_I$ Estimation

We have seen that we can express any movement by means of $WCC_I$. Computing $WCC_I(v, C, P)$ requires computing the triangles $v$ and those vertices in $C$ close with the other vertices in $C$ and $v$. For a vertex, this operation is bound by the number of neighbors that have $(d)$, which has a complexity of $O(d^2)$ (for each neighbor in $C \cup \{v\}$ we have to test against all of its other neighbors in $C \cup \{v\}$). Also, since real graphs typically have power law distributions, this cost is large for the highest degree vertices in the graph. Finally, considering that the number of times $WCC_I$ is called is bounded by the number of edges $m$ of the graph, it quickly becomes the most time consuming part of the algorithm. In this section, we propose a model to estimate $WCC_I()$ with a constant time complexity function (given some easy to compute statistics) that we call $WCC'_I()$.

$WCC'_I()$ stands as the approximated increment of $WCC$ when vertex $v$ is inserted into a community $C$. In Figure 10, we depict the simplified model on top of which $WCC'_I()$ is built. For a given vertex $v$, we only record the number of edges that connect it to community $C$. For each community $C$, we keep the following statistics: the size of the community $r$; the edge density of the community $\delta$; and the number of edges $b$ that are in the boundary of the community. We also use the clustering coefficient of the graph $\omega$, which is constant along all the community detection process and has been computed during the “Graph cleanup” step. The clustering coefficient of the graph is equivalent as the observed probability that two given edges that share a vertex close a triangle. These statistics homogenize the community members and allow the computation of $WCC'_I()$ as follows:

THEOREM 7. *Consider the situation depicted in Figure 10, with the following assumptions:*

— *Every edge in the graph closes at least one triangle.*
— *The edge density inside community $C$ is homogeneous and equal to $\delta$ .*
— *The clustering coefficient of the whole graph equals to $\omega$.*

*Then,*

$$\begin{aligned} WCC(P') - WCC(P) &= WCC'_I(v, C) \\ &= \frac{1}{V} \cdot (d_{in} \cdot \Theta_1 + (r - d_{in}) \cdot \Theta_2 + \Theta_3), \end{aligned} \tag{7}$$

*where,*

$$\begin{aligned}\Theta_1 &= \frac{(r-1)\delta+1+q}{(r+q)\cdot((r-1)(r-2)\delta^3+(d_{in}-1)\delta+q(q-1)\delta\omega+q(q-1)\omega+d_{out}\omega)}\cdot (d_{in}-1)\delta;\\ \Theta_2 &= -\frac{(r-1)(r-2)\delta^3}{(r-1)(r-2)\delta^3+q(q-1)\omega+q(r-1)\delta\omega}\cdot\frac{(r-1)\delta+q}{(r+q)(r-1+q)};\\ \Theta_3 &= \frac{d_{in}(d_{in}-1)\delta}{d_{in}(d_{in}-1)\delta+d_{out}(d_{out}-1)\omega+d_{out}d_{in}\omega}\cdot\frac{d_{in}+d_{out}}{r+d_{cout}};\end{aligned}$$

*and* $q = (b - d_{in})/r$.

Conceptually, $\Theta_1$, $\Theta_2$ and $\Theta_3$ are the $WCC$ improvements of those vertices in $C$ connected to $v$, those vertices in $C$ not connected to $v$, and vertex $v$ respectively, when $v$ is added to community $C$. The evaluation of Equation 7 is $O(1)$ given all the statistics. And, the update of all statistics is only performed when all communities are updated, with a cost $O(m)$ for the whole graph. Note that we use aggregated statistics to estimate the number of triangles, and thus we are *not* computing the triangles when we compute $WCC'_I()$.

### 6.5. Complexity of the Algorithm

Let $n$ be the number of vertices and $m$ the number of edges in the graph. We assume that the average degree of the graph is $d = m/n$ and that real graphs have a quasi-linear relation between vertices and edges $O(m) = O(n \cdot \log n)$. Then, the complexity of each of the steps of the algorithm is the following:

**Graph Cleanup:** In the graph cleanup phase, for each edge in the graph, we compute the triangles that each edge participates in. The triangles are found by intersecting the adjacency lists of the two connected vertices. Since we assume sorted adjacency lists, the complexity of computing the intersection is $O(d)$. Finally, we compute the local clustering coefficient of each vertex and and the number of triangles each vertex closes, which has a cost of $O(m)$ once we have the triangles each edge participates in. Since the average degree is $m/n$, we have that the cost of the first phase is $O(m \cdot d + m) = O(m \cdot \log n + m)$.

**Initial Partition:** The cost of this step is the cost of sorting the vertices of the graph based on the local clustering coefficients computed in the previous phase, which is $O(n \cdot log(n))$.

**Partition Refinement:** Let $\alpha$ be the number of iterations required to find the best partition P', which in our experiments is between 3 and 7. In each iteration, for each vertex $v$ of the graph, we compute, in the worst case, $d + 1$ movements of type $WCC'(I)$ that have a cost $O(1)$. Then, the computation of the best movement for all vertices in the graph in an iteration is $O(n \cdot (d + 1)) = O(m)$. The application of the all the movements is linear with respect to the number of vertices $O(n)$. We also need to update, for each iteration of the second phase, the statistics $\delta$, $c_{out}$, $d_{in}$ and $d_{out}$ for each vertex and community, which has a cost of $O(m)$. Finally, the computation of the $WCC$ for the current partition is performed by computing for each edge the triangles, which is $O(m \cdot \log n)$ as already stated. Hence, the cost of the refinement phase becomes $O(\alpha \cdot (m + n + m + m \cdot \log n))$, which after simplification, becomes $O(m \cdot \log n)$ assuming $\alpha$ as constant.

Finally, The final cost of the algorithm is the sum of the three phases: $O(m \cdot \log n + n \cdot log(n) + m \cdot \log n) = O(m \cdot \log n)$.

## 7. DETECTABILITY ANALYSIS OF $WCC$

Recent studies have revealed the difficulties of existing community detection metrics, such as modularity, to detect communities if they are not well defined [Decelle et al. 2011]. Typically, these studies use simplified graph models, being the *Stochastic Block Model* one of the most widely used. This model assumes a graph with $q$ communities, where there is an edge between two vertices with probability $p_{in}$ if both belong to the same community, and probability $p_{out}$ if they belong to different communities. The question is whether a given community detection metric is able to detect the communities of the model for a given set of configuration parameters ($p_{in}$, $p_{out}$, $n$, and $q$). In this section we analyze the level of detectability of $WCC$ using the stochastic block model. For the sake of simplicity, in our study we will stick to the case where we have a graph with n vertices, consisting of $q = 2$ communities of size $\frac{n}{2}$.

Given a stochastic block model graph $G$ of size $n$ and two communities of size $\frac{n}{2}$, we want to find the *detectability threshold* of $WCC$, that is, the point in the relation between $p_{in}$ and $p_{out}$ where maximizing $WCC$ obtains the expected communities of the model. Actually, this can be also seen in terms of *intraconnectivity* and *isolation*: the value of $p_{in}$ models the *intraconnectivity* of the communities of the model (the larger $p_{in}$, the better the *intraconnectivity*) while $p_{out}$ models the *isolation* of the communities (the lower $p_{out}$ is, the better the *isolation*). The *detectability threshold* of $WCC$ is defined in Theorem 8, whose proof can be found in Appendix F.

THEOREM 8. *Let* G *be a an arbitrarily large graph with* $n$ *vertices with two communities* $A$ *and* $B$ *of size* $\frac{n}{2}$ *each. Let* $C(x)$ *be the community where vertex* $x$ *is assigned. Two vertices* $x$ *and* $y$ *are connected with probability* $p_{in}$ *if* $C(x) = C(y)$, *and with probability* $p_{out}$ *if* $C(x) \neq C(y)$. *Let* $\mathcal{P}$ *be any possible partition of the graph, being* $\mathcal{P}_1 = \{A, B\}$ *and* $\mathcal{P}_2 = \{A \cup B\}$ *particular instances of that partition. Then:*

(i)

$$WCC(\mathcal{P}_1) = \frac{(\frac{n}{2}-1)(\frac{n}{2}-2)\cdot p_{in}^3 \cdot ((\frac{n}{2}-1)\cdot p_{in} + \frac{n}{2}\cdot p_{out})}{((\frac{n}{2}-1)(\frac{n}{2}-2)\cdot p_{in}^3 + (\frac{n}{2}-1)n\cdot p_{in}\cdot p_{out}^2)(\frac{n}{2}\cdot p_{out} + \frac{n}{2} - 1)}$$

(ii)

$$WCC(\mathcal{P}_2) = \frac{((\frac{n}{2}-1)\cdot p_{in} + \frac{n}{2}\cdot p_{out})}{n-1}$$

(iii) $\arg\max_{\mathcal{P}} WCC(\mathcal{P}) \in \{\mathcal{P}_1, \mathcal{P}_2\}$ *if and only if* $p_{in} > p_{out}$*;*

(iv) $WCC(\mathcal{P}_1) > WCC(\mathcal{P}_2)$ *if and only if*

$$p_{in} > \frac{\sqrt{(2-2\cdot p_{out})(p_{out}+1)}\cdot p_{out}}{1-p_{out}} \tag{8}$$

Theorem 8 states that the partition with optimal $WCC$ for these particular graphs is either that containing the original communities ($\{A, B\}$) or that taking the graph as a single community ($\{A \cup B\}$), and the transition point between the former and the later, is that expressed by Equation 8. In other words, $WCC$ is able to recover the original communities if and only if the condition in Equation 8 (the *detectability threshold*) holds.

### 7.1. $WCC$'s detectability discussion

The question is whether the detectability threshold of $WCC$ is good or not. Intuitively, according to the informal community definition, one would expect a community detection metric to detect communities whenever $p_{in} > p_{out}$. However, this definition is incomplete and, in practice, $p_{in} > p_{out}$ is not a sufficient condition to define a community. As an example, suppose a clique of size $n$. Just removing an edge of the clique would imply a configuration with two communities of size $\frac{n}{2}$, with $p_{in} = 1 > p_{out} = \frac{n^2-4}{n^2}$, if we strictly adhere to the informal community definition. Therefore, a community metric with a detectability

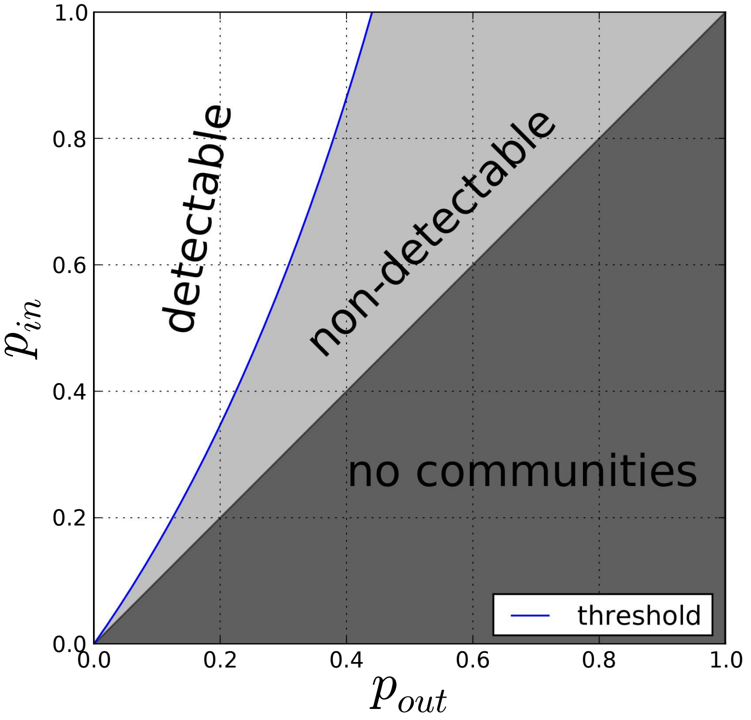


Fig. 11. Transition point for $WCC$ and different values of $p_{in}$ and $p_{out}$

threshold of $p_{in} > p_{out}$ would potentially only detect communities with a perfectly uniform edge distribution. Otherwise, these would break up into smaller subcommunities with a uniform density. Clearly, this is not practical in a real application, as real graphs are typically not uniform and situations such as that described above appear frequently. Therefore, the actual transition point between detectable and non-detectable configurations for a given metric is a direct consequence of the community definition of that metric, and whether this is good or bad is determined by the application in use. In the case of $WCC$, the application is social networks and, as shown in Section 9, $SCD$, the algorithm proposed in Section 6, outperforms the current state of the art algorithms in this same domain.

Besides this, several desirable properties are observed from the detectability threshold of $WCC$, that are good indicators of the behavior of the metric. First, it is independent of the size of the graph, as already proven in Property 2. This means that the balance between *intraconnectivity* and *isolation* for a given configuration, is constant regardless of the size of the community. This is not the case, for instance, for modularity, whose detectability threshold is $c_{in} - c_{out} \geq 2\sqrt{c_{in} + c_{out}}$, where $c_{in} = \frac{n}{2} \cdot p_{in}$ and $c_{out} = \frac{n}{2} \cdot p_{out}$. In this case, the smaller the graph, the larger $c_{in}$ needs to be compared to $c_{out}$ to be able to correctly identify the communities. This issue is related to the known resolution problems of modularity, which is unable to detect small and well defined communities once the size of the graph increases.

Second, the implicit community definition proposed by $WCC$ is not static, that is, it does not impose either a minimum level of *intraconnectivity* or *isolation* to a set of vertices

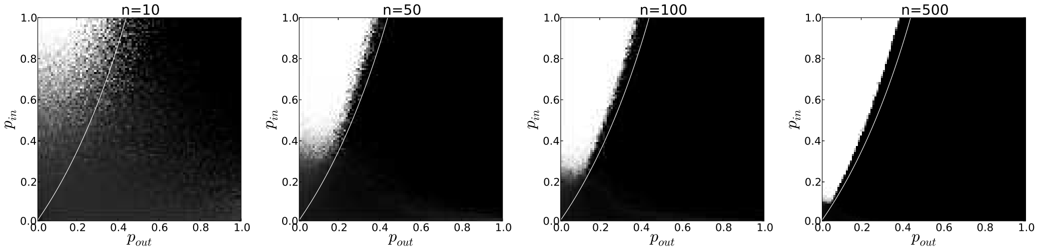


Fig. 12. Detectability of the proposed algorithm for stochastic block model graphs of different sizes with different configuration parameters ($p_{in}$ and $p_{out}$). The closer to white the color is, the better the NMI between the detected partition and that expected by the model. The closer to black, the more different.

to be detected as a community. This is better observed in Figure 11, where we plot three well defined regions, corresponding to the different configuration spaces where $WCC$ is able to detect the communities, where it is not able to detect them and where communities do not exist ($p_{in} \leq p_{out}$). We also show the detectability threshold of $WCC$, which delimits the detectable and non-detectable regions. We see that the threshold establishes a relation between $p_{in}$ (*intraconnectivity*) and $p_{out}$ (*isolation*), in such a way that the more intraconnected the communities are (the larger $p_{in}$ is), the less isolated (the larger $p_{out}$) these can be and vice versa. This means that $WCC$ is a metric that is not static but locally identifies relevant sets of vertices, thus adapting to the heterogeneous nature of real graphs. Finally, we see that $WCC$ does not detect the original communities whenever they do not exist – i.e. when $p_{in} \leq p_{out}$ –, which is desirable in a community detection metric not to detect false positives.

In Figure 12 we show a numerical validation of the detectability threshold of $WCC$. We test different configurations of stochastic block model graphs ($p_{in}$ and $p_{out}$ from 0 to 1 in steps of 0.01, for different values of $n$) with two communities using the $WCC$ maximization algorithm proposed in [Prat-Pérez et al. 2014]. We generated 100 graphs for each tested configuration, executed the algorithm, and compared the resulting partition with that expected from the model using the *Normalized Mutual Information* (NMI) [Fortunato 2010]. Each point in the graph corresponds to the average NMI of those 10 executions. The whiter the color, the closer to one the average NMI is (the algorithm finds the expected communities), and the darker the color, the closer to zero the NMI is (the communities found by the algorithm are very different from those expected from the model). We also draw the detectability threshold.

We see the detectability threshold line fits very well with the empirical results obtained, with a very well defined transition point between the detectable and non-detectable regions. The larger the size of the graph, the better this fitness, because the larger the graph, and as a consequence, the larger the communities, the more uniform the internal edge density of these is. Furthermore, we also empirically confirm that when communities do not exist ($p_{in} \leq p_{out}$), $WCC$ does not detect them. These empirical validation also suggests that the algorithm proposed in [Prat-Pérez et al. 2014] is able to produce results close to the optimal, even though it does not formally guarantee to produce an optimal solution.

### 7.2. The community detection paradox and $WCC$

Another issue that affects modularity maximization based algorithms is the so called community detection paradox [Radicchi 2014]. Counterintuitively, the paradox states that the worse defined the communities are, the easier it is for the algorithms to detect them, while the better defined they are, the harder it is. In order to test $WCC$ is not affected by this issue or not, we first need to define what is a good and a bad community in terms of $WCC$, and then, test whether in situations where we have bad communities, they are harder or easier

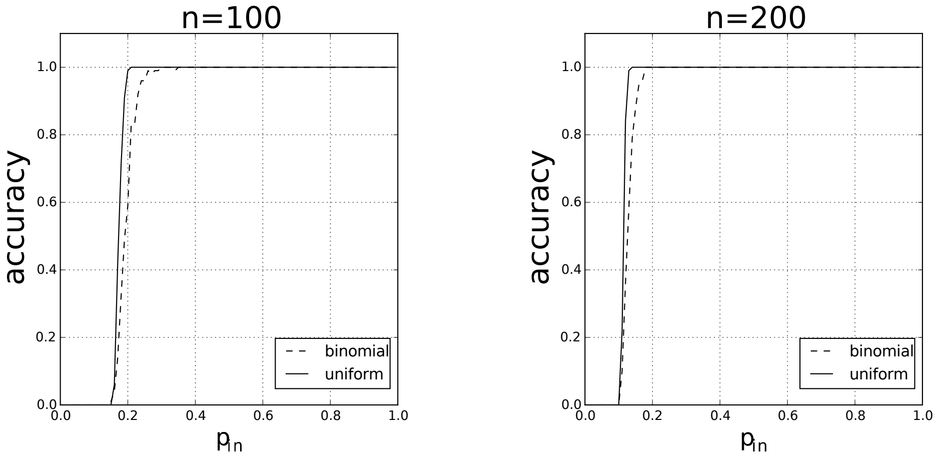


Fig. 13. Accuracy of $SCD$ to detect a community with a uniform vs binomial degree distribution, for different expected values of $p_{in}$.

to detect. More concretely, for $WCC$, a well defined community shows both a good isolation and intraconnectivity. Also, a bad community is not well isolated nor intraconnected[4].

We first analyze the case where a community is perfectly isolated, that is, it does not have external edges connecting its vertices to other communities. The *detectability threshold* shows that when the internal degree of the vertices of the community is uniform, the larger the $p_{in}$ is, the larger the $p_{out}$ can be and the community can still be detectable up to a certain point. In the case that the degree of the vertices is not uniform, some vertices might be well intraconnected, while others might not close enough triangles to be part of the community. If this happens, the community structure is not so well defined and thus, we expect $WCC$ to fail at identifying the community.

Figure 13 shows the accuracy $SCD$, when the degrees of the vertices follow a *Binomial* distribution compared to when the degrees of the vertices are uniform, for different values of $p_{in}$ and communities of size $n = 100$ and $n = 200$ vertices. In this case the graph consists of a single community, thus $p_{out}$ is zero. For each configuration, we have randomly generated 100 graphs, executed the algorithm and averaged the results. An accuracy of one means that the algorithm is able to fully recover the communities for all of the 100 generated instances, while an accuracy of zero means that it was not able to recover the communities in some of them. We see that as long as the value of $p_{in}$ increases, there is a transition point for both distributions where the algorithm starts to correctly detect the community for all the instances of the graph. This transition point is seen earlier for the uniform graph than for the binomial. In the case of $WCC$, one would expect this transition point not to exist for perfectly uniform graphs and always detect the communities, as predicted in Theorem 1 where the number of edges required between a vertex and a community tends to zero as $p_{in}$ approaches zero. However, this is the case for arbitrarily large graphs, but not for small graphs where having perfectly uniformly distributed edges is more difficult as these either exist or not. Therefore, the smaller $p_{in}$ is, the larger the probability a vertex not having enough edges with the community or even not closing any triangle with it exists. This probability is larger for the case of the binomial distribution, where degrees are less homogeneous.

[4] Note, that in the case of modularity and other state of the art algorithms these definitions change, and a community is usually considered to be well defined when its vertices have more internal edges than external edges.

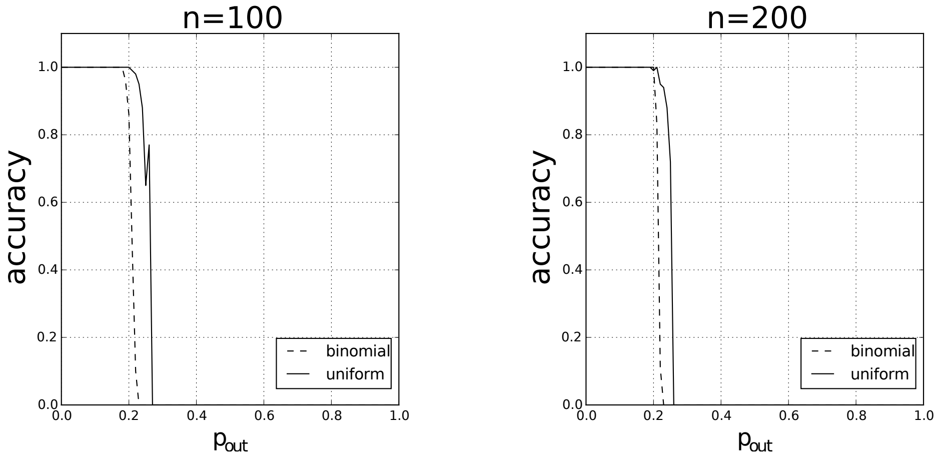


Fig. 14. Accuracy of $SCD$ to detect two communities with a uniform degree distribution with density $p_{in} = 0.5$, and a uniform vs binomial out degree distribution, for different expected values of $p_{out}$.

We also tested the situation where we have two well intraconnected and uniform communities, but the out degree distribution connecting both communities is not uniform but follows a Binomial distribution. In this case, we expect that for the Binomial distribution the communities will be harder to detect, as some observed vertices will have a larger out degree than that expected, making them less isolated from the rest of the graph, even sometimes being more intraconnected with vertices of the other community. Figure 14 shows the accuracy of the community detection algorithm based on $WCC$ optimization, when the two communities have a uniform distributed internal degree with $p_{in}$ of 0.5, and the out degree follows a Binomial and a uniform degree distribution, for different values of $p_{out}$. In this case, we see that when the out degree follows a Binomial distribution, the transition point between detectable configurations and non-detectable configurations is seen earlier (for smaller values of $p_{out}$).

In conclusion, we see that $WCC$ is sensitive to how the edges are internally and externally distributed. The more uniformly distributed these are, the easiest is for $WCC$ to detect the communities. If some vertices do not have enough intraconnection or isolation, then $WCC$ will not classify them in the correct community, as expected.

## 8. EXPERIMENTAL SETUP

We select the most relevant algorithms in the state of the art in order to detect the communities in real world graphs. Then, we prove that there is a correlation between $WCC_s$ and different statistical indicators, that is, communities with good $WCC_s$ have good statistical indicators, while communities with bad $WCC_s$ fail at one or more of those statistical indicators. Finally, we compare the existing algorithms using ground truth communities and $SCD$, and show that not only $SCD$ obtains the best results and performs better, but also there is a strong correlation between $WCC$ and the quality of the communities.

The selected algorithms of the state of the art are *Infomap* [Rosvall and Bergstrom 2008], which is based on random walks; *Louvain* [Blondel et al. 2008], which is based on multilevel maximization of modularity locally; *Label Propagation Method (LPM)* [Raghavan and Albert 2007], which finds communities using a label propagation approach. We choose Infomap and Louvain because they are the best for detecting disjoint communities in social networks according to [Lancichinetti 2009], and LPM because it has become very popular in the literature due to its linear time complexity. We suggest reading the reference paper for each algorithm to understand them in detail. Our selection covers algorithms following

Table I. Real-world graphs with ground truth data.

| | Vertices | Edges | Communities |
|---|---|---|---|
| Amazon | 334,863 | 925,872 | 151,037 |
| Dblp | 317,080 | 1,049,866 | 13,477 |
| Youtube | 1,134,890 | 2,987,624 | 8,385 |
| LiveJournal | 3,997,962 | 34,681,189 | 287,512 |
| Orkut | 3,072,441 | 117,185,083 | 6,288,363 |
| Friendster | 65,608,366 | 1,806,067,135 | 957,154 |

diverse strategies to test the validity of $WCC$ but it does not intend to be an evaluation survey of all community methods. Besides, other popular approaches in the literature, such as [Lancichinetti et al. 2011; Ahn et al. 2010; Palla et al. 2005] among others, aim at overlapping communities which are also out of the scope of this paper. The implementation of all the algorithms has been taken from their author's web site.

For the experimentation, we used six real networks covering different aspects of real world data, mostly social networks[5]. All chosen networks have ground truth communities associated with them. The first is a network representing which products from Amazon have been copurchased by clients. In this dataset, the ground truth communities match the different categories of products. The second is a graph of the DBLP network representing coautorship relations between authors, where ground truth communities correspond to authors that have published in the same journals and conferences. The third graph is a graph of Youtube, where ground truth communities correspond to the groups of users in youtube. The fourth , fifth and sixth datasets are graphs of the Livejournal, Orkut and Friendster social networks, where ground truth communities correspond to the groups created by the users. The characteristics of these graphs are summarized in Table I.

Finally, we used a machine with the following characteristics: 2xIntel Xeon E5-2609 @ 2.40GHz, with 4 cores each making a total of 8 cores, 128 GB ram and Linux 2.6.32-5-amd64. The used disks are regular 1TB spinning disks at 7200 rpm.

## 9. EXPERIMENTAL RESULTS

### 9.1. Statistical Indicators analysis

In this section, we show the correlation between communities with good $WCC$ values and good statistical indicators. As statistical indicators, we have taken a selection of existing metrics described in Section 2, that include representatives of each of the different categories of metrics. Furthermore, we added a new category composed of structural properties. The formal definitions can be found in Appendix K.

**Internal connectivity based indicators (I)**: We use the *average edge density*, the *triangle density*, and the *TPR* to measure the level of *intraconnectivity* of the communities.

**External connectivity based indicators (E)**: We use the *expansion* [Radicchi et al. 2004] to measure level of *isolation* of the communities.

**Internal and external connectivity based metrics (I+E)**: We use the *average inverse edge cut*, which is the average fraction of internal edges out of the total number of edges of a vertex, the *Flake ODF* [Flake et al. 2000] and the *conductance* [Kannan et al. 2004], as a tradeoff between *intraconnectivity* and *isolation.*

**Graph model comparison based indicators (M)**: We use the *modularity* per community [Newman and Girvan 2004], because it measures how relevant is a community, and because of its popularity in the literature.

[5] Downloaded from SNAP (http://snap.stanford.edu). We cleaned the original graphs by removing the self loops.

**Structural indicators (S)**: We use the *bridge ratio*, the *normalized diameter* and the *size* of the communities, to obtain a better insight into the characteristics of the communities.

We created a pool of communities by running *Infomap*, *Louvain* and *LPM* on the first four real world networks described above[6]. We sorted all the communities in the pool by their $WCC_s$ value decreasingly although they have not been found with such metric. Then, we divided the communities into 20 groups in steps of five percentiles according to their $WCC_s$ and plotted for these 20 groups their corresponding statistical indicators in Figure 15. In all the charts, the $x$ axis represents the group identifier (e.g. the leftmost group is always the 95 percentile that contains the top 5% communities in terms of their $WCC_s$) while the $y$ axis shows the corresponding statistical value. The communities of size one and two, are omitted since their $WCC_s$ value is always zero. As shown in Figure 15(a), the leftmost communities have high $WCC_s$ values, and the rightmost communities have the lowest $WCC_s$ values. Since these communities were not computed with $WCC_s$, we analyze both good and bad communities.

Broadly speaking, we observe two sections in each plot of Figure 15: from groups 1 to 11, the trend for all statistical indicators show that communities with higher $WCC_s$ have better properties; from groups 12 to 20 this trend apparently changes in some statistical indicators. We focus first on groups 1-11 and we analyze groups 12-20 later.

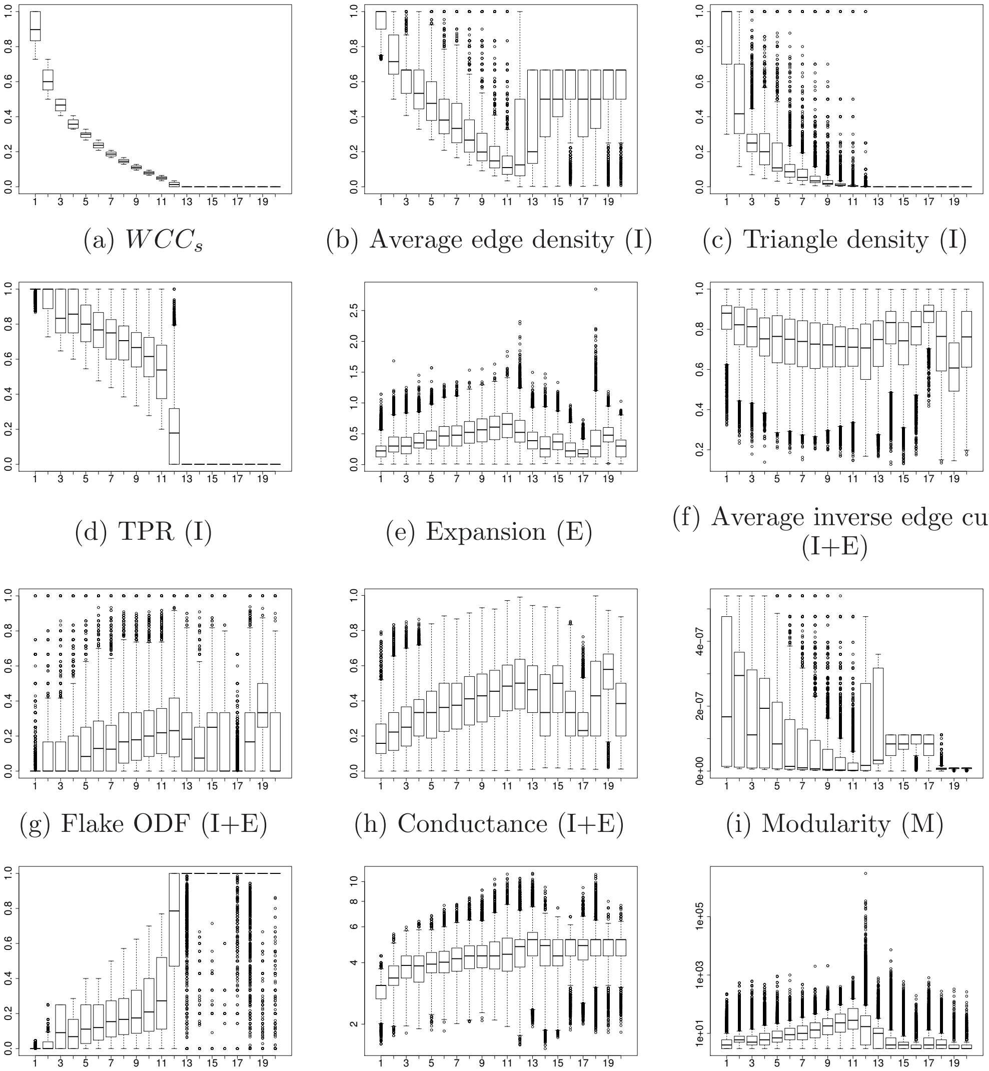
*Groups 1-11:* In Figures 15(b) and (c) we see that the larger the $WCC_s$ of a community, the larger the average edge density and the triangle density. The transitive relations between the vertices (Property 1) indicate the presence of communities with a defined homophilic structure. Note that these communities have been found with metrics that do not search for triangles and yet, they contain more such structures. Similarly, Figure 15(d) shows that the larger the $WCC_s$, the larger the TPR. However, we see that TPR has precision limitations, since it scores as good communities (with scores close to one), some communities that might not be that good according to other metrics such as the average edge density (Figure 15(b)), the expansion (Figure 15(e)), the average inverse edge cut (Figure 15(f)) and the conductance (Figure 15(g)). Finally, in Figures 15(e), (f) and (g), we see that the larger the $WCC_s$, the smaller the expansion, the larger the average inverse edge cut, and the smaller the conductance, which means that the number of external connections decreases for the first, and that the communities are denser internally than externally for the last two. However, while having a large internal density and few external connections is a good starting point to identify a good community, it does not imply an internal structure as we will show when discussing groups 12-20.

In Figure 15(h-i) we compare $WCC_s$ with the most used metrics in the state of the art: conductance and modularity. We see that for these groups, there is a correlation between communities with good $WCC_s$ values, modularity and conductance (note that for conductance, the lower, the better).

Figure 15(j) shows that bridges penalize the $WCC_s$ score. A large bridge ratio is a symptom of the presence of whiskers or treelike structures, which are inherently sparse and hence do not have the type of internal structure that one would expect from a community. A small diameter is another feature that any good community should have. In Figure 15(k) we see that large $WCC_s$ implies smaller diameters for the communities. This means that any vertex in the community is close to any other vertex, which translates to denser communities. Finally, in Figure 15(l), we show the sizes of the communities.

*Groups 12-20:* We see that there is a trend change in some statistical indicators for those groups that have $WCC_s$ close to 0. This behavior can be explained by Figures 15(c), (d) and (j). These figures reveal that the communities after group 15 are treelike: communities hardly contain triangles and almost all the edges in the community are bridges. Such structures

[6] We used the first four as they where the only ones where all the algorithms succeded to execute



(a) $WCC_s$ (b) Average edge density (I) (c) Triangle density (I)

(d) TPR (I) (e) Expansion (E) (f) Average inverse edge cut (I+E)

(g) Flake ODF (I+E) (h) Conductance (I+E) (i) Modularity (M)

(j) Bridge ratio (S) (k) Normalized diameter (S) (l) Size (S)

Fig. 15. Statistics of communities from real world networks in 20 groups sorted by $WCC_s$. The x-axis represents the 5% percentile groups showing that with the largest $WCC$ on the left and that with the smallest $WCC$ on the right. The y-axis represents the value achieved for each of the metrics shown in the plots

cannot be accepted as good communities. Although some communities in groups 15-20 are isolated, we note that this is not a sufficient condition for them to be good communities. For example, most communities in groups 13-20 are trees with three vertices which have a good conductance. $WCC_s$ is able to score these communities as bad communities while conductance does not. A similar behavior is seen for modularity. In Figure 15(i), we see that the communities in groups 13-20, which are tree-like, have a larger modularity than other sets with a more community-like structure than those. As described in [Bagrow 2012], tree

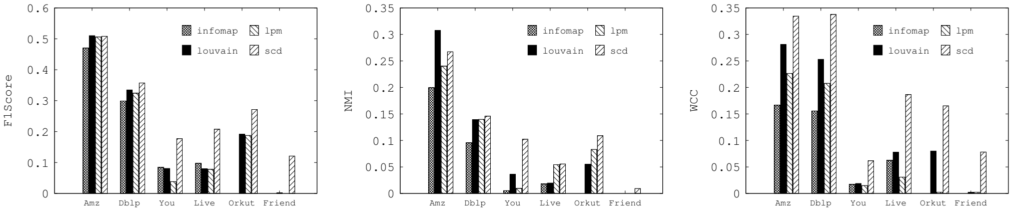


Fig. 16. F1Score, NMI and $WCC$.

like networks can have high modularity and hence, algorithms maximizing it can lead to misleading results.

To sum up, while state of the art metrics fail to correctly rank communities under specific circumstances, $WCC_s$ shows to be robust, that is, it is able to globally capture all the desirable characteristics a community should contain.

### 9.2. Evaluating $WCC$ using Ground Truth

In the previous section, we correlated $WCC_s$ with a set of statistical indicators which show the presence of a community structure. In this section, we use ground truth communities to correlate $WCC$ with those communities found in real data. With this objective, we test the quality of the partitions obtained by the algorithms on the different datasets and compare them to the available ground truth communities. Inspired by the methodology of [Yang and Leskovec 2013], the quality of a partition with respect to a ground truth is measured using the **Average F1Score** ($\bar{F_1}$) and NMI. The F1Score of a set $A$ with respect to a set $B$ is defined as the harmonic mean ($H$) of the precision and the recall of $A$ with respect to $B$:

$$\begin{aligned} precision(A,B) &= \frac{|A \cap B|}{|A|},\ recall(A,B) = \frac{|A \cap B|}{|B|}. \\ H(a,b) &= \frac{2 \cdot a \cdot b}{a+b} \\ F_1(A,B) &= H(precision(A,B), recall(A,B)) \end{aligned}$$

Then, the average F1Score of two sets of communities $C_1$ and $C_2$ (which in our case are the partition and the ground truth communities respectively) is given by:

$$\begin{aligned} F_1(A,C) &= \arg\max_i F_1(A,C_i),\ c_i \in C = \{C_1, \cdots, C_n\} \\ \bar{F_1}(C_1,C_2) &= \frac{1}{2|C|}\sum_{c_i \in C} F_1(c_i, C') + \frac{1}{2|C'|}\sum_{c_i \in C'} F_1(c_i, C) \end{aligned}$$

We also compare the quality of the results obtained using the *Normalized Mutual Information* (NMI), which is widely used in the community detection literature [Fortunato 2010].

Figure 16 shows the Average F1Score, NMI and $WCC$ of the partition obtained by the different algorithms, in the tested graphs. Those missing bars are from executions that were not able to finish within a week or consumed too much memory. We observe that there is a strong correlation between $WCC$, and the F1Score and NMI obtained, that is, in general, the larger the $WCC$, the better the F1Score and NMI obtained.

In order to quantify the correlation between F1Score, NMI and $WCC$, we computed the Pearson Coefficient of variation that resulted **0.91** and **0.83** for F1Score and NMI respectively. This indicates a very strong agreement between both metrics and $WCC$ since it is close to 1, which is the maximum value. Therefore, $WCC$ proves to be a solid metric for evaluating the quality of community detection algorithms.

### 9.3. SCD Performance and Scalability

In Figure 17(a) we show the excution times of the different algorithms single threaded, for the different graphs. We see that SCD is the fastest algorithm for the smaller graphs, and the second fastest after LPM and Louvain when these become larger. However we see that the execution times are still competitieve compared with the implementations of the state of the art algorithms used. Again, those missing bars belong to those executed that were unable to finish in less than a week or due to memory consumption.

We parallelized SCD in order to exploit the resources of current multi-core and many-core processors. More concretely, we parallelized the two most time consuming parts of the algorithm: the computation of the global and local clustering coefficient of the vertices during the graph clean up phase and the whole refinement phase. In the former, we parallelized the loop that computes, for each edge, the number of triangles that the edge closes. In the later, we parallelized both the loop in Line 7 of Algorithm 2, which calls the function `bestMovement` for each of the vertices in the graph and the computation of $WCC$ for the partition at the end of the iteration (which can be paralelized for each vertex). Since all the parallel code is in the form of loops, we used OpenMP with dynamic scheduling, using a chunk size of 32. Figure 17(b) shows the normalized execution times of SCD with different number of threads. In this experiment, we have excluded the time spent in I/O, which includes reading the graph file and printing the results.

Broadly speaking, we see that with a simple OpenMP based parallelization, SCD is able to achieve very good scalability, specially for the larger graphs which are also those with a larger average degree. The larger the average degree of the graph, the larger the cost of those parts that have been parallelized: the larger the cost of computing $WCC$ and the larger the number of movements to test between vertices and communities). These two parts quickly become dominant over the sequential ones. This means a better scalability due to a direct application of Ahmdal's Law.

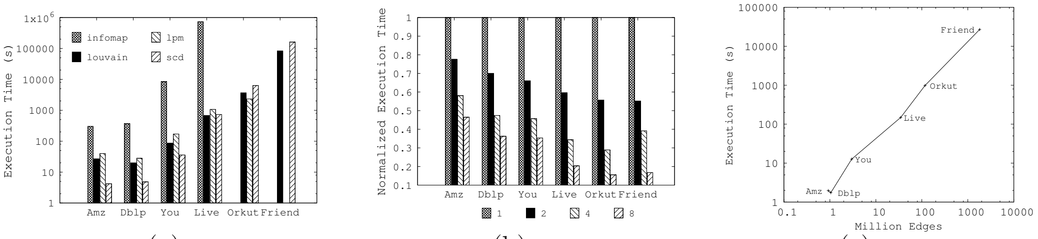
We see then that for large graphs, our implementation of SCD is able to exploit all the processor's resources available. The configuration with eight threads of SCD keeps the eight cores of the processor active most of the time, obtaining between six and seven fold improvement over the single threaded version. These results show that SCD is an algorithm easy to parallelize and capable of exploiting multi-core architectures efficiently, especially on those cases (large graphs) where this is more appreciated. More specifically, SCD processes the Friendster graph using eight threads in just 7.5 hours.

Figure 17(c) shows the execution time of SCD with respect to the number of edges of the graph. Each point represents the time spent by the eight thread version of SCD for the different graphs. We see that SCD shows a quasi linear scalability, as described in Section 6.5.

### 9.4. Memory Consumption

In Table II, we show the memory consumption in MB of SCD for each of the graphs divided into three categories:

- **Graph**: the size of the data structure that stores the graph as a list of adjacencies. The graph is stored in compressed sparse row format. We relabel the original ids of the vertices to the range from 0 to n-1, and hence, we also account an array containing a mapping between our internal vertex identifier and the original label used in the input files.
- **Triangles**: an array with size equals to the number of vertices, which contains the number of triangles each vertex belongs to.



(a) (b) (c)

Fig. 17. (a) Execution times of the different algorithms single threaded. (b) SCD normalized execution time with different number of threads. (c) Execution time with eight threads vs number of edges.

Table II. SCD Memory consumption in MB.

| | Graph | Triangles | Partitions | Total |
|---|---|---|---|---|
| Amazon | 11.4 | 1.3 | 16.0 | 28.7 |
| Dblp | 12.2 | 1.3 | 14.9 | 28.4 |
| Youtube | 37.5 | 4.5 | 68.9 | 110.9 |
| Livejournal | 325.4 | 16.0 | 197.7 | 539.1 |
| Orkut | 974.3 | 12.3 | 124.4 | 1111.0 |
| Friendster | 15235.8 | 262.4 | 3317.6 | 18815.8 |

— **Partitions**: accounts for the partition and its statistics. In this case, we report the iteration with the largest memory consumption.

We see that the data structures (array of triangles and partitions) built by SCD scale linearly with the number of vertices of the graph, and not with the number of edges. Furthermore, since the number of statistics we store per vertex is small, the amount of memory consumed by SCD is often dominated by the graph representation as its memory consumption depends on the number of edges of the graph. This is observed by all the tested graphs except Youtube, which has a very small average degree of 2.6. For the largest graphs, SCD allocates only data structures for an additional 23% 23% (Friendster) and 14% (Orkut) of the original graph. The amount of memory consumed for the Friendster graph is roughly 18GB, showing that much larger graphs could be processed with a comodity server with the 128 GB of memory.

### 9.5. Use case

Finally, we have created a graph of journals, where the vertices are computer science journals and where two journals are connected if they are similar in terms of topics (they share authors, which are expected to work on similar topics) and size (have a similar number of publishing authors). In order to construct the graph, we have used the DBLP data available on their site [Dblp 2012]. This graph is actually a social graph, because the interests of the authors and their collaborations are driven by the homophilic principle that has been discussed all over this paper. We create an edge if the Jaccard Coefficient between the lists of authors (the authors of those papers published in the journal) of two journals is above a given threshold. This means that two journals are related, if the set of authors that have published in both journals is large with respect to the total number of authors that have published in one of the journals. We set the threshold at 0.04 (this number has been taken after several experiments in order to obtain a representative graph).

Table III shows four examples of communities found by the use of $WCC$ in $SCD$ on this graph, related to the Knowledge Discovery , Databases, Graphics and Neuroscience topics. We see that each community, is formed by journals of its topic, meaning that maximizing $WCC$ allows extracting meaningful communities.

Table III. Examples of communities of journals found from maximizing $WCC$.

| | |
|---|---|
| Data Mining | Data Mining and Knowledge Management; Intell. Data Analysis; Knowledge Information Systems; SIGKDD Explorations; Statistical Analysis and Data Mining; TKDD. |
| Data Management | ACM Trans. Database Syst.; Data Knowl. Eng.; Distributed and Parallel Databases; IEEE Data Eng. Bull.; IEEE Trans. Knowl. Data Eng.; Inf. Syst.; Int. J. Cooperative Inf. Syst.; J. Intell. Inf. Syst.; PVLDB; SIGMOD Record; VLDB J.; World Wide Web. |
| Graphics | ACM Trans. Graph.; Comput. Graph. Forum; Computer Aided Geometric Design; Computer-Aided Design; Graphical Models; Graphics; IEEE Computer Graphics and Applications; IEEE Trans. Vis. Comput. Graph.; Journal of Visualization and Computer Animation The Visual Computer. |
| Neuroscience | Biological Cybernetics; IEEE Trans. Neural Netw. Learning Syst.; IEEE Transactions on Neural Networks; Int. J. Neural Syst.; Journal of Computational Neuroscience; Neural Computation; Neural Computing and Applications; Neural Networks; Neural Processing Letters; Neurocomputing. |

## 10. CONCLUSIONS AND FUTURE WORK

Although different community metrics have been proposed, these do not guarantee cohesive and structured communities from their optimization. Actually, the most popular metrics applied in the state of the art fail at correctly ranking the quality of a community under certain circumstances. The reason is that existing metrics do not consider the internal structure of the community, but focus only on maximizing/minimizing global characteristics. We observed that community detection metrics should fulfill a minimal set of properties, guaranteeing communities with a minimal level of structure.

In this paper, we proposed $WCC$, a new community detection metric that quantifies the quality of a community. By meeting the above mentioned properties, $WCC$ is able to guarantee that the communities delivered from its optimization will be cohesive and structured. We have shown experimentally that communities with a good $WCC$ are dense, have small edge cuts, have transitive relations without bridges and small diameters. We have also shown that looking only at the internal density and small edge cuts does not guarantee well defined communities with internal structure, since it can lead to treelike communities.

We have also analyzed the detectability threshold of $WCC$, and shown that it fulfills a set of desirable properties, such as that it is independent of the size of the communities, or that $WCC$ does not find communities when these do not exist.

We have correlated $WCC$ with a set of statistical indicators, showing that communities with a large $WCC$, have the characteristics expected from good communities. We have empirically shown that $WCC$ is a metric correlated with real graph communities, and hence serve as a good baseline to compare community detection algorithms when ground truth data is not available.

Along with the metric, we have proposed a *Scalable Community Detection*$(SCD)$, a community detection algorithm based on $WCC$ optimization. We have designed $SCD$ with parallelism in mind, and have shown its performance and scalability in practice using real graphs.

In this paper, we have discussed the concepts of structural isolation and intraconnectivity. The importance of these concepts is treated equally from the $WCC$'s perspective. However, some applications would prefer a metric more biased to isolation, while others would consider more important the intraconnectivity. Future work will consist on adding a mechanism to adjust the importance or weight of such concepts inside the $WCC$ definition. This will allow $WCC$ to be more flexible and to be adapted to a wider range of applications. Another interesting problem is the search of overlapped communities in the graph. Some graph patterns, such as cut vertices, can be naturally modeled as the overlap of two communities.

This problem, similarly to the disjoint case, has similar deficiencies because there is not a formalization of a minimal set of properties to be fulfilled by a metric. Our work will continue towards extending the community definition and $WCC$ for overlapping communities.

## 11. ACKNOWLEDGMENTS

The members of DAMA-UPC thank the Ministry of Science and Innovation of Spain and Generalitat de Catalunya, for grants TIN2013-47008-R and GRC-2014-890 respectively, and IBM CAS Canada Research for their strategic research grant, and Oracle for its graphalytics project grant. Brunat thanks the Ministry of Science and Innovation of Spain and Generalitat de Catalunya for grants MTM2011-24097 and DGR-2009SGR1040 respectively. Dominguez-Sal thanks the Ministry of Science and Innovation of Spain for the grant Torres Quevedo PTQ-11-04970.

## REFERENCES

Y.Y. Ahn, J.P. Bagrow, and S. Lehmann. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466, 7307 (2010), 761–764.

J. P. Bagrow. 2012. Are communities just bottlenecks? Trees and treelike networks have high modularity. *CoRR* abs/1201.0745 (2012).

V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. 2008. Fast unfolding of communities in large networks. *JSTAT* 2008 (2008), P10008.

A. Clauset, M.E.J. Newman, and C. Moore. 2004. Finding community structure in very large networks. *Phy. Rev. E* 70, 6 (2004), 066111.

Dblp. 2012. http://dblp.uni-trier.de/xml/. (Sept. 2012). http://dblp.uni-trier.de/xml/

Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. 2011. Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters* 107, 6 (2011), 065701.

E. Di Giacomo, W. Didimo, L. Grilli, and G. Liotta. 2007. Graph visualization techniques for web clustering engines. *TVCG* (2007), 294–304.

G.W. Flake, S. Lawrence, and C.L. Giles. 2000. Efficient identification of web communities. In *SIGKDD*. ACM, 150–160.

S. Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3-5 (2010), 75–174.

S. Fortunato and M. Barthélemy. 2007. Resolution limit in community detection. *PNAS* 104, 1 (2007), 36.

M. Girvan and M.E.J. Newman. 2002. Community structure in social and biological networks. *PNAS* 99, 12 (2002), 7821.

B. H. Good, Y.A. de Montjoye, and A. Clauset. 2010. Performance of modularity maximization in practical contexts. *Phy. Rev. E* 81, 4 (2010), 046106.

Joan Guisado-Gámez, David Dominguez-Sal, and Josep-Lluis Larriba-Pey. 2014. Massive Query Expansion by Exploiting Graph Knowledge Bases for Image Retrieval. In *International Conference on Multimedia Retrieval, ICMR '14, Glasgow, United Kingdom - April 01 - 04, 2014*. 33.

R. Kannan, S. Vempala, and A. Vetta. 2004. On clusterings: Good, bad and spectral. *JACM* 51, 3 (2004), 497–515.

A. Lancichinetti. 2009. Community detection algorithms: a comparative analysis. *Phy. Rev. E* 80, 5 (2009), 056117.

A. Lancichinetti, F. Radicchi, J.J. Ramasco, and S. Fortunato. 2011. Finding statistically significant communities in networks. *PloS one* 6, 4 (2011), e18961.

J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. 2008. Microscopic evolution of social networks. In *SIGKDD*. ACM, 462–470.

J. Leskovec, K. J. Lang, and M. Mahoney. 2010. Empirical comparison of algorithms for network community detection. In *WWW*. ACM, 631–640.

D. Liben-Nowell and J. Kleinberg. 2007. The link-prediction problem for social networks. *ASIST* 58, 7 (2007), 1019–1031.

M. McPherson, L. Smith-Lovin, and J. M. Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* (2001), 415–444.

A. Medus, G. Acuna, and CO. Dorso. 2005. Detection of community structures in networks via global optimization. *Physica A* 358, 2-4 (2005), 593–604.

M.E.J. Newman. 2001. The structure of scientific collaboration networks. *PNAS* 98, 2 (2001), 404.

M.E.J. Newman and M. Girvan. 2004. Finding and evaluating community structure in networks. *Phy. Rev. E* 69, 2 (2004), 026113.

M.E.J. Newman and J. Park. 2003. Why social networks are different from other types of networks. *Phy. Rev. E* 68, 3 (2003), 036122.

A. Padrol-Sureda, G. Perarnau-Llobet, J. Pfeifle, and V. Muntés-Mulero. 2010. Overlapping Community Search for social networks. In *ICDE*. 992–995.

G. Palla, I. Derényi, I. Farkas, and T. Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435, 7043 (2005), 814–818.

A. Prat-Pérez, D. Dominguez-Sal, and J.L. Larriba-Pey. 2011. Social Based Layouts for the Increase of Locality in Graph Operations. In *DASFAA*. 558–569.

A. Prat-Pérez, D. Dominguez-Sal, and J. L. Larriba-Pey. 2014. High Quality, Scalable and Parallel Community Detection for Large Real Graphs. *To be published in WWW* (2014).

Filippo Radicchi. 2014. A paradox in community detection. *EPL (Europhysics Letters)* 106, 3 (2014), 38001.

F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. 2004. Defining and identifying communities in networks. *PNAS* 101, 9 (2004), 2658.

U. N. Raghavan and S. Albert, Rmara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76, 3 (2007), 036106.

M. Rosvall and C.T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *PNAS* 105, 4 (2008), 1118.

V. Satuluri, S. Parthasarathy, and Y. Ruan. 2011. Local graph sparsification for scalable clustering. In *SIGMOD*. ACM, 721–732.

X. Shi, L.A. Adamic, and M.J. Strauss. 2007. Networks of strong ties. *Physica A: Statistical Mechanics and its Applications* 378, 1 (2007), 33–47.

M. Sozio and A. Gionis. 2010. The community-search problem and how to plan a successful cocktail party. In *KDD*. 939–948.

T. Wang, B. Yang, J. Gao, D. Yang, S. Tang, H. Wu, K. Liu, and J. Pei. 2009. Mobileminer: a real world case study of data mining in mobile communication. In *SIGMOD*. ACM, 1083–1086.

J. Yang and J. Leskovec. 2012. Defining and Evaluating Network Communities Based on Ground-Truth. In *ICDM*. 745–754.

J. Yang and J. Leskovec. 2013. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM*. 587–596.

Jaewon Yang and Jure Leskovec. 2014. Overlapping Communities Explain Core-Periphery Organization of Networks. *Proc. IEEE* 102, 12 (2014), 1892–1902.

## A. PROOF OF PROPOSITION 3.1

PROOF. (i) This is a consequence of the inequalities $t(x,S) \leq t(x,V)$ and $vt(x,S) \leq vt(x,V)$, $vt(x,V) \leq vt(x,V) + |S \setminus \{x\}| - vt(x,S)$

(ii) If $WCC_v(x,S) = 0$, then at least one of the following three identities holds: $t(x,V) = 0$, $vt(x,V) = 0$, and $t(x,S) = 0$. Now, each one of these conditions implies $t(x,S) = 0$. Reciprocally, by definition, if $t(x,S) = 0$, then $WCC_v(x,S) = 0$.

(iii) Assume $WCC_v(x,S) = 1$. By (ii), $t(x,S) \neq 0$. Hence, there exists an edge $\{y,z\}$ such that $y \in S \setminus \{x\}$ and $z \in S \setminus \{x\}$ forming triangle with $x$. Then $|S \setminus \{x\}| \geq 2$. As the two fractions defining $WCC_v(x,S)$ are $\leq 1$, the condition $WCC_v(x,S) = 1$ implies that both fractions are 1. Left fraction is 1 if and only if $t(x,V) = t(x,S)$, which implies that $vt(x,V) \leq |S \setminus \{x\}|$, which turns into an equality (and therefore right fractions becomes 1) if and only if $vt(x,S) = |S \setminus \{x\}|$.

Reciprocally, the condition $vt(x,V) = |S \setminus \{x\}| = vt(x,S) \geq 2$ implies $t(x,S) = t(x,V)$. As $vt(x,V) = vt(x,S) = |S \setminus \{x\} \geq 2$, we have that both fractions in the definition of $WCC_v(x,S)$ have denominator $\neq 0$ and both fractions are 1. Therefore, $WCC_v(x,S) = 1$. □

## B. PROOF OF PROPOSITION 3.2

PROOF. The proofs are a consequence of Proposition 3.1. (i) Since $0 \leq WCC_v(x,S) \leq 1$ for all $x \in S$, then $0 \leq WCC_s(S) \leq 1$.

(ii) $WCC_s(S) = 0$ implies that for all $x \in S$ $WCC_v(x,S) = 0$. Since the condition for $WCC_v(x,S) = 0$ is that $t(x,S) = 0$, then $WCC_s(S) = 0$ implies that $S$ has no triangles.

(iii) $WCC_s(S) = 1$ implies that $WCC_v(x,S) = 1$ for all $x \in S$. This implies that a vertex $x \in S$ such that $vt(x,V) \neq vt(x,S)$ and $vt(x,S) = |S \setminus \{x\}|$ does not exist. Thus, all the vertices $x \in S$ form triangles only and with exactly all the vertices in $S$, which implies having an edge with all the vertices in $S$, and hence forming a clique. □

## C. PROOF OF THEOREM 1

PROOF. Let $N$ be the set of neighbors of $v$.

(i) For $x \in V$, we have $WCC_v(x,V) = vt(x,V)/r$. Now,

$$vt(x,V) = \begin{cases} (r-1)p & \text{if } x \in V \setminus N; \\ (r-1)p+1 & \text{if } x \in N; \\ d & \text{if } x \in \{v\}. \end{cases}$$

Then

$$\begin{aligned}(r+1)WCC(\mathcal{P}_1) =& (r-d)\frac{(r-1)p}{r} + d\frac{(r-1)p+1}{r} + \frac{d}{r} \\ =& (r-1)p + 2\frac{d}{r}.\end{aligned}$$

(ii) For $x \in V \setminus N$,

$$WCC_v(x,V) = \frac{vt(x,V)}{r-1} = \frac{(r-1)p}{r-1} = p.$$

(iii) For $x \in N$, we have

$$\begin{aligned} t(x,V) &= \binom{r-1}{2}p^3; \\ t(x,V\cup\{v\}) &= \binom{r-1}{2}p^3 + (d-1)p; \\ vt(x,V\cup\{v\}) &= (r-1)p+1; \\ |V \setminus \{x\}| - vt(x,V) &= (r-1)-(r-1)p; \end{aligned}$$

$$WCC_v(x,V) = \frac{((r-1)p+1)(r-1)(r-2)p^2}{((r-1)(r-2)p^2 + 2(d-1))\cdot r}.$$

Moreover, $WCC_v(v, \{v\}) = 0$. Then,

$$(r+1)WCC(\mathcal{P}_2) = (r-d)p + \frac{d}{r} \cdot \frac{((r-1)p+1)(r-1)(r-2)p^2}{(r-1)(r-2)p^2 + 2(d-1)}.$$

(iv) We have,

$$(r+1)\left(WCC(\mathcal{P}_1) - WCC(\mathcal{P}_2)\right) = p(d-1) + 2\frac{d}{r} - \frac{d}{r}\frac{((r-1)p+1)(r-1)(r-2)p^2}{(r-1)(r-2)p^2+2(d-1)},$$

and the condition $WCC(\mathcal{P}_1) - WCC(\mathcal{P}_2) > 0$ is equivalent to the condition

$$ad^2 + bd + c > 0, \tag{9}$$

where

$$\begin{aligned} a =& 2(2+pr), \\ b =& p^2(p+1)r^2 - p(3p^2+3p+4)r + 2p^3 + 2p^2 - 4, \\ c =& -p^3r^3 + 3p^3r^2 + 2p(1-p^2)r. \end{aligned}$$

For short, let we denote by $O(r^n)$ a polynomial expression of degree at most $n$. Then, the greatest solution of (9) is,

$$d_2 = \frac{-p^2(1+p)r^2 + O(r) + \sqrt{p^4(p^2+2p+9)r^4 + O(r^3)}}{4(2+pr)}$$

and we get

$$\begin{aligned} \lim_{r\to+\infty} \frac{d_2}{r} &= \frac{-p^2(1+p) + p^2\sqrt{p^2+2p+9}}{4p} \\ &= p\frac{\sqrt{p^2+2p+9} - (1+p)}{4}. \end{aligned}$$

Thus, for a large enough $r$, the condition

$$d > rp\left(\sqrt{p^2+2p+9} - (1+p)\right)/4,$$

is equivalent to $WCC(\mathcal{P}_1) > WCC(\mathcal{P}_2)$. □

Note that function $p \mapsto p\left(\sqrt{p^2+2p+9} - (1+p)\right)/4$ is increasing in $p$. A large value of $p$ means more edges in $G$, and then a large value of $d/r$ is needed for $WCC(\mathcal{P}_1)$ being greater than $WCC(\mathcal{P}_2)$.

In the case of Corollary 1, $p = 1$, thus $d > \sqrt{3} - 1/2 = 0.37$.

## D. PROOF OF THEOREM 2

Proof. Let $S = S_1 \cup S_2$. For $x \in S_i, i \in \{1,2\}$ we have $t(x, S_i) = t(x, S)$, $vt(x, V \setminus S_i) = vt(x, V \setminus S)$ and $|S_i \setminus \{x\}| < |S \setminus \{x\}|$. Then,

$$WCC_v(x,S) \;=\; \frac{t(x,S)}{t(x,V)} \cdot \frac{vt(x,V)}{vt(x,V) + |S \setminus \{x\}| - vt(x,S)} < \frac{t(x,S_i)}{t(x,V)} \cdot \frac{vt(x,V)}{vt(x,V) + |S_i \setminus \{x\}| - vt(x,S_i)} = WCC_v(x,S_i).$$

Therefore,

$$\begin{aligned} |S| \cdot WCC(\{S_1,S_2\}) &= |S_1| \cdot WCC_s(S_1) + |S_2| \cdot WCC_s(S_2) \\ &= \sum_{x\in S_1} WCC_v(x,S_1) + \sum_{x \in S_2} WCC_v(x,S_2) > \sum_{x\in S} WCC_v(x,S) \end{aligned}$$

implies

$$WCC(\{S_1,S_2\}) > \frac{1}{|S|}\sum_{x\in S} WCC_v(x,S) = WCC_s(S) = WCC_s(S_1 \cup S_2).$$

□

## E. PROOF OF THEOREM 3

PROOF. (i) For the $r-1$ vertices $x \in K_r \setminus \{t\}$, we have $WCC_v(x,V) = vt(x,V)/(n-1) = (r-1)/(n-1)$. For the vertex $t$, we have $WCC_v(v,V) = 1$. Finally, for the $s-1$ vertices $x \in K_s \setminus \{t\}$, we have $WCC_v(x,V) = (s-1)/(n-1)$. As $n-1 = r+s-2$, we obtain the formula (4).

(ii) For the $r-1$ vertices $x \in K_r$ we have $WCC_v(x,K_r) = 1$. For the vertex $t$, we have

$$\begin{aligned} WCC_v(x,K_r) =& \frac{\binom{r-1}{2}}{\binom{r-1}{2}+\binom{s-1}{2}} \cdot \frac{n-1}{r-1+s-1} \\ =& \frac{(r-1)(r-2)}{(r-1)(r-2)+(s-1)(s-2)}. \end{aligned}$$

For the $s-1$ vertices $x \in K_s \setminus \{t\}$, we have

$$WCC_v(x,K_s \setminus \{t\}) = \frac{\binom{s-2}{2}}{\binom{s-1}{2}} \cdot \frac{s-1}{s-1} = \frac{(s-2)(s-3)}{(s-1)(s-2)}.$$

This gives the formula (5).

(iii) For $x \in K_r \setminus \{t\}$,

$$WCC_v(x,K_r \setminus \{t\}) = \frac{\binom{r-2}{2}}{\binom{r-1}{2}} \cdot \frac{r-1}{r-1} = \frac{(r-2)(r-3)}{(r-1)(r-2)};$$

for vertex $t$,

$$\begin{aligned} WCC_v(x,\{t\}) =& \frac{\binom{0}{2}}{\binom{n-1}{2}} \cdot \frac{n-1}{0+r-1+s-1} \\ =& \frac{(1-1)(1-2)}{(r+s-2)(r+s-3)} = 0; \end{aligned}$$

for $x \in K_s \setminus \{t\}$,

$$WCC_v(x,K_s \setminus \{t\}) = \frac{(s-2)(s-3)}{(s-1)(s-2)};$$

This implies (6).

(iv) Define $f_1(r,s) = n \cdot WCC(\mathcal{P}_1)$, $f_2(r,s) = n \cdot WCC(\mathcal{P}_2)$, and $f_3(r,s) = n \cdot WCC(\mathcal{P}_3)$. The expression of these functions are those in (4), (5) and (6), respectively. The goal is to show that for all integers values $r$, $s$ with $r \geq s \geq 4$ the inequality $f_3(r,s) \leq f_2(r,s)$ holds. Clearly, the first summand of $f_3(r,s)$ is smaller than the first summand of $f_2(r,s)$, and the last summands are equal. As $f_2(r,s)$ has the second summand $\geq 0$, we have $f_3(r,s) \leq f_2(r,s)$.

(v) We shall prove $f_2(r,s) - f_1(r,s) \geq 0$ for $n \geq 7$ and $4 \leq r \leq n-3$. We have $s = n-r+1$ and

$$\begin{aligned} f_2(r,s) - f_1(r,s) >& n-4-\frac{(r-1)^2+(n-r)^2}{n-1} \\ =& \frac{-2r^2+(2+2n)r-5n+3}{n-1}. \end{aligned}$$

The sign of $f_2(r,s) - f_1(r,s)$ is the sign of the polynomial function $-2r^2+2(n+1)r-5n+3$, which is a convex function on $r$ with roots:

$$\begin{aligned} r_1 =& \frac{1}{2}(n+1-\sqrt{n^2-8n+7}); \\ r_2 =& \frac{1}{2}(n+1+\sqrt{n^2-8n+7}). \end{aligned}$$

Now, for $n \geq 7$, we have $r_1 \leq 4$ and $r_2 \geq n-3$. Therefore, for each $r \in \{4,\ldots,n-3\}$ we have $f_2(r,s) - f_1(r,s) \geq 0$. □

## F. PROOF OF THEOREM 8

PROOF. Let $x$ be any vertex of the graph $G(V,E)$, and $S$ the community of vertex $x$. Let's assume that all the edges of the graph close at least one triangle.
(i) For $\mathcal{P}_1$

$$\begin{aligned}
t(x,S) &= \binom{\frac{n}{2}-1}{2} p_{in}^3;\\
t(x,V) &= \binom{\frac{n}{2}-1}{2} p_{in}^3 + \binom{\frac{n}{2}}{2} p_{in}\cdot p_{out}^2 + \binom{\frac{n}{2}-1}{1}\binom{\frac{n}{2}}{1} p_{in}\cdot p_{out}^2;\\
vt(x,V) &= (\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out};\\
vt(x,V)+|S\setminus\{x\}|-vt(x,S) &= (\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out}\frac{n}{2} - 1 - (\frac{n}{2}-1)p_{in}.
\end{aligned}$$

Then,

$$WCC(\mathcal{P}_1) = \frac{(\frac{n}{2}-1)(\frac{n}{2}-2)p_{in}^3((\frac{n}{2}-1)p_{in}+\frac{n}{2}p_{out})}{((\frac{n}{2}-1)(\frac{n}{2}-2)p_{in}^3 + (\frac{n}{2}-1)n\cdot p_{in}\cdot p_{out}^2)(\frac{n}{2}p_{out}+\frac{n}{2}-1)}.$$

(ii) For $\mathcal{P}_2$

$$\begin{aligned}
t(x,S) &= \binom{\frac{n}{2}-1}{2} p_{in}^3 + \binom{\frac{n}{2}}{2} p_{in}\cdot p_{out}^2 + \binom{\frac{n}{2}-1}{1}\binom{\frac{n}{2}}{1} p_{in}\cdot p_{out}^2;\\
t(x,V) &= \binom{\frac{n}{2}-1}{2} p_{in}^3 + \binom{\frac{n}{2}}{2} p_{in}\cdot p_{out}^2 + \binom{\frac{n}{2}-1}{1}\binom{\frac{n}{2}}{1} p_{in}\cdot p_{out}^2;\\
vt(x,V) &= (\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out};\\
vt(x,V)+|S\setminus\{x\}|-vt(x,S) &= n-1.
\end{aligned}$$

Then,

$$WCC(\mathcal{P}_2) = \frac{(\frac{n}{2}-1)p_{in}+\frac{n}{2}p_{out}}{n-1}.$$

(iii) We numerically proof this statement. First, we need to compute the $WCC$ of $\mathcal{P}_s$, which consist of those partitions where $s$ vertices of each of the two communities have been correctly placed. More formally, let $\mathcal{P}_s = \{A', B'\}$ be any partition of the graph with two communities $A'$ and $B'$ of size $\frac{n}{2}$, where $|A\cap A'| = s$ and $|B\cap B'| = s$. Let $x_a \in \{A\cap A'\}$, $x_b \in \{B\cap B'\}$, $v_a \in \{A'\setminus A\}$ and $v_b \in \{B'\setminus B\}$. That is, any partition with two communities of size $\frac{n}{2}$ where $2s$ vertices have been well placed.

$$\begin{aligned}
t(x_a, A') &= \binom{s-1}{2} p_{in}^3 + \binom{s-1}{1}\binom{\frac{n}{2}-s}{1} p_{in} \cdot pout^2 + \binom{\frac{n}{2}-s}{2} p_{in} \cdot pout^2; \\
t(x_b, B') &= t(x_a, A'); \\
t(x_a, V) &= \binom{\frac{n}{2}-1}{2} p_{in}^3 + \binom{\frac{n}{2}}{2} p_{in} \cdot p_{out}^2 + \binom{\frac{n}{2}-1}{1}\binom{\frac{n}{2}}{1} p_{in} \cdot p_{out}^2; \\
t(x_b, V) &= t(x_a, V); \\
vt(x_a, V) &= (\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out}; \\
vt(x_b, V) &= vt(x_a, V); \\
vt(x_b, V) + |A' \setminus \{x_a\}| - vt(x_a, A') &= (\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out} + (\frac{n}{2} - 1 - (s-1)p_{in} - (\frac{n}{2}-s)p_{out}); \\
vt(x_b, V) + |B' \setminus \{x_b\}| - vt(x_b, B') &= vt(x_b, V) + |A' \setminus \{x_a\}| - vt(x_a, A').
\end{aligned}$$

$$\begin{aligned}
t(v_a, A') &= \binom{\frac{n}{2}-s-1}{2} p_{in}^3 + \binom{\frac{n}{2}-s-1}{1}\binom{s}{1} p_{in} \cdot pout^2 + \binom{s}{2} p_{in} \cdot pout^2; \\
t(v_b, B') &= t(v_a, A'); \\
t(v_a, V) &= \binom{\frac{n}{2}-1}{2} p_{in}^3 + \binom{\frac{n}{2}}{2} p_{in} \cdot p_{out}^2 + \binom{\frac{n}{2}-1}{1}\binom{\frac{n}{2}}{1} p_{in} \cdot p_{out}^2; \\
t(v_b, V) &= t(v_a, V); \\
vt(v_a, V) &= (\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out}; \\
vt(v_b, V) &= vt(v_a, V); \\
vt(v_b, V) + |A' \setminus \{v_a\}| - vt(v_a, A') &= (\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out} + (\frac{n}{2} - 1 - (s-1)p_{in} - (\frac{n}{2}-s)p_{out}); \\
vt(v_b, V) + |B' \setminus \{v_b\}| - vt(v_b, B') &= vt(v_b, V) + |A' \setminus \{v_a\}| - vt(v_a, A').
\end{aligned}$$

Then,

$$WCC(\mathcal{P}_s) = \frac{1}{n} \cdot 2 \cdot s \cdot \frac{(s-1)(s-2)p_{in}^3 + (s-1)(\frac{n}{2}-s)p_{in} \cdot p_{out}^2 + (\frac{n}{2}-s)(\frac{n}{2}-s-1)p_{out}^2 \cdot p_{in}}{((\frac{n}{2}-1)(\frac{n}{2}-2)p_{in}^3 + (\frac{n}{2}-1)n \cdot p_{in} \cdot p_{out}^2)} \cdot \frac{(\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out}}{(\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out} + (\frac{n}{2} - 1 - (s-1)p_{in} - (\frac{n}{2}-s)p_{out})}$$

$$+$$

$$\frac{1}{n} \cdot 2 \cdot (\frac{n}{2}-s) \frac{((\frac{n}{2}-s)-1)((\frac{n}{2}-s)-2)p_{in}^3 + ((\frac{n}{2}-s)-1)s \cdot p_{in} \cdot p_{out}^2 + s(s-1)p_{out}^2 \cdot p_{in}}{((\frac{n}{2}-1)(\frac{n}{2}-2)p_{in}^3 + (\frac{n}{2}-1)n \cdot p_{in} \cdot p_{out}^2)} \cdot \frac{(\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out}}{(\frac{n}{2}-1)p_{in} + \frac{n}{2}p_{out} + (\frac{n}{2} - 1 - (\frac{n}{2}-s-1)p_{in} - s \cdot p_{out})}.$$

Figure 18 shows, for each configuration of $p_{in}$ and $p_{out}$, for different values of $n$, which partition $\mathcal{P} \in \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_s\}$ is that with a maximum $WCC$. In the case of $\mathcal{P}_s$, we tested for all possible values of $s$. We see that the statement is true, regardless of the the value of $n$.

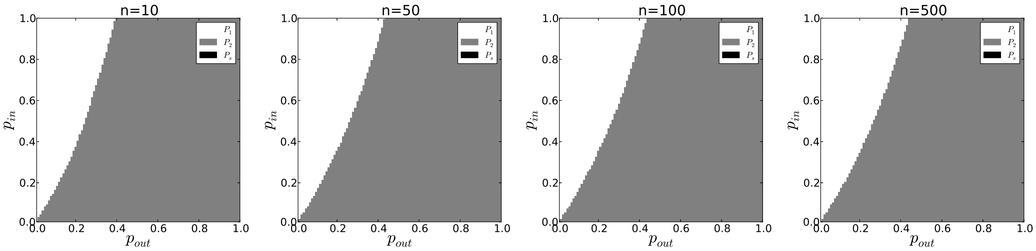


Fig. 18. The best partition found for different configurations of $p_{in}$, $p_{out}$ and $n$. All possible configurations of $\mathcal{P}_s$ have been tested

(iv) We are interested in the transition point between $\mathcal{P}_1$ and $\mathcal{P}_2$, that is, we want to know for which values of $p_{in}$ and $p_{out}$, $WCC(\mathcal{P}_1) - WCC(\mathcal{P}_2) = 0$. Since we are interested in arbitrarily large graphs, we compute the difference when $n$ tends to infinitely large values :

$$\mathcal{L} = \lim_{n\to\infty} WCC(\mathcal{P}_1) - WCC(\mathcal{P}_2) = -\frac{1}{2}\frac{(p_{in}^2 \cdot p_{out} + 2 \cdot p_{out}^3 - p_{in}^2 + 2 \cdot p_{out}^2)(p_{in} + p_{out})}{(p_{out}+1)(p_{in}^2 + 2 \cdot p_{out}^2)}$$

If we solve $\mathcal{L} = 0$ for $p_{in}$, we obtain the following solutions:

$$-p_{out}, -\frac{\sqrt{(2-2\cdot p_{out})(p_{out}+1)}p_{out}}{1-p_{out}}, \frac{\sqrt{(2-2\cdot p_{out})(p_{out}+1)}p_{out}}{1-p_{out}}$$

, being the third solution the only positive and valid one. □

## G. PROOF OF THEOREM 4

PROOF.

$$\begin{aligned}
&WCC(P') - WCC(P) = \\
&={}^1\!/_{|V|}\left(|C_1 \cup \{v\}| \cdot WCC(C_1 \cup \{v\}) + \sum\nolimits_{i=2}^{k} |C_i| \cdot WCC(C_i)\right) - \\
&{}^1\!/_{|V|}\left(|C_1| \cdot WCC(C_1) + \sum\nolimits_{i=2}^{k} |C_i| \cdot WCC(C_i) + WCC(\{v\})\right) \\
&={}^1\!/_{|V|}(|C_1'| \cdot WCC(C_1')) - {}^1\!/_{|V|}(|C_1| \cdot WCC(C_1) + 0)
\end{aligned}$$

$$\begin{aligned}
&={}^1\!/_{|V|}\left(\sum\nolimits_{x\in C_1'} WCC(x, C_1') + \sum\nolimits_{x\in C_1} WCC(x, C_1)\right) \\
&={}^1\!/_{|V|}\left(\sum\nolimits_{x\in C_1} WCC(x, C_1') + WCC(v, C_1') - \right. \\
&\left. \sum\nolimits_{x\in C_1} WCC(x, C_1)\right)
\end{aligned}$$

□

## H. PROOF OF THEOREM 5

PROOF. As stated in the theorem assumptions, the partition $P'$ is build by removing $v$ from $C_1$. Alternatively, the partition $P$ can be build by removing vertex $v$ to $C_1'$ in $P'$. Then, the two following equalities hold:

$$\begin{aligned}
WCC(P) + WCC_R(v, C_1) &= WCC(P'), \\
WCC(P) &= WCC(P') + WCC_I(v, C_1') \\
\text{and thus: } WCC_R(v, C_1) &= -WCC_I(v, C_1')
\end{aligned}$$

□

## I. PROOF OF THEOREM 6

PROOF. Since $WCC$ is a state function, all paths from $P$ to $P'$ have the same differential. Then, we express the transfer operation as a combination of remove and insert:

$$WCC(P) + WCC_T(v, C_1, C_k) = WCC(P')$$
$$WCC(P) + WCC_R(v, C_1) + WCC_I(v, C_k) = WCC(P')$$
$$WCC(P') - WCC(P) = -WCC_I(v, C_1') + WCC_I(v, C_k)$$

□

## J. PROOF OF THEOREM 7

PROOF. Consider the situation depicted in Figure 10. Let $N(x)$ be the set of neighbors of $x$. Given that, we define sets $F = N(v) \cap C$ which contains those vertices in $C$ that are actual neighbors of $v$, and $G = (C \setminus N(x))$, which contains those vertices in $C$ that are not neighbors of $v$. Therefore, from Theorem 4 we have:

$$\begin{aligned}
&WCC_I(v, C) = \\
&\quad = {}^1\!/\!_{|V|} \sum\nolimits_{x \in C} (WCC(x, C \cup \{v\}) - WCC(x, C)) + \\
&\quad\ {}^1\!/\!_{|V|} WCC(v, C \cup \{v\}) \\
&\quad = {}^1\!/\!_{|V|} \sum\nolimits_{x \in F} (WCC(x, C \cup \{v\}) - WCC(x, C)) + \\
&\quad\ {}^1\!/\!_{|V|} \sum\nolimits_{x \in G} (WCC(x, C \cup \{v\}) - WCC(x, C)) + \\
&\quad\ {}^1\!/\!_{|V|} WCC(v, C \cup \{v\})
\end{aligned}$$

We know that $|F| = d_{in}$ and $|G| = r - d_{in}$, then we can define $WCC_I'(v, C)$ with respect to three variables $\Theta_1$, $\Theta_2$ and $\Theta_3$, which represent the $WCC$ improvement of a vertex of $F$, a vertex of $G$ and $v$ respectively. Then,

$$WCC_I'(v, C) = {}^1\!/\!_{|V|} (|F| \cdot \Theta_1 + |G| \cdot \Theta_2 + \Theta_3).$$

We define $q = (b - d_{in})/r$ as the number of edges connecting each vertex in $C$ with the rest of the graph excluding $v$. Then,

(i) If $x \in F$, we have

$$\begin{aligned}
t(x, C) =& (r-1)(r-2)\delta^3; \\
t(x, C \cup \{v\}) =& (r-1)(r-2)\delta^3 + (d_{in} - 1)\delta; \\
t(x, V) =& (r-1)(r-2)\delta^3 + (d_{in} - 1)\delta + q(r-1)\delta\omega + \\
& q(q-1)\omega + d_{out}\omega; \\
vt(x, V) =& (r-1)\delta + 1 + q;
\end{aligned}$$

$$\begin{aligned}
vt(x, V) + |C \cup \{v\} \setminus \{x\}| - vt(x, \{C \cup \{v\}\}) = r + q; \\
vt(x, V) + |C \setminus \{x\}| - vt(x, C) = r - 1 + q + 1 = r + q;
\end{aligned}$$

In $t(x, C)$, we account for those triangles that $x$ closes with two other vertices in $C$. Similarly, in $t(x, C \cup \{v\})$ we account for those triangles that $x$ closes with two other vertices in $C$, and those triangles that $x$ closes with $v$ and another vertex in $C$. $t(x, V)$ accounts for all triangles that vertex $x$ closes with the graph, which are: $t(x, C \cup \{v\})$ plus those triangles that vertex $x$ closes with another vertex of $C$ and a vertex of $V \setminus C$, plus those triangles that vertex $x$ closes with two other vertices in $V \setminus C$, plus those triangles vertex $x$ closes with $v$ and another vertex of $V \setminus C$. Since we assume that every edge in the graph closes at least one triangle, $vt(x, V)$ accounts for the number of vertices in $C$ that are actual neighbors of $x$ plus 1 (for vertex $v$) and $q$ vertices that are connected to $x$. Finally, we have that the union of vertices in $C$ and those vertices in $V$ with whom $x$ closes at least one triangle is $r + q$. Therefore,

$$\begin{aligned}\Theta_1 = & \quad WCC(x, C \cup \{v\}) - WCC(x, C)\\ = & \quad \tfrac{t(x,C\cup\{v\})}{t(x,V)} \cdot \tfrac{vt(x,V)}{|C\cup\{v\}\setminus\{x\}|+vt(x,V\setminus\{C\cup\{v\}\})} - \\ & \quad \tfrac{t(x,C)}{t(x,V)} \cdot \tfrac{vt(x,V)}{|C\setminus\{x\}|+vt(x,V\setminus C)}\\ = & \quad \tfrac{vt(x,V)}{(r+q)\cdot t(x,V)} \cdot (t(x, C \cup \{v\}) - t(x, C))\\ = & \quad \tfrac{(r-1)\delta+1+q}{(r+q)\cdot((r-1)(r-2)\delta^3+(d_{in}-1)\delta+q(r-1)\delta\omega+q(q-1)\omega+d_{out}\omega)} \cdot \\ & \quad (d_{in}-1)\delta.\end{aligned}$$

(ii) If $x \in B$, we have

$$\begin{aligned}t(x, C) =&(r-1)(r-2)\delta^3;\\ t(x, C \cup \{v\}) =&(r-1)(r-2)\delta^3;\\ t(x, V) =&(r-1)(r-2)\delta^3 + q(q-1)\omega + q(r-1)\delta\omega;\\ vt(x, V) =&(r-1)\delta + q;\end{aligned}$$

$$\begin{aligned}&vt(x, V) + |C \cup \{v\} \setminus \{x\}| - vt(x, \{C \cup \{v\}\}) = r + q;\\ &vt(x, V) + |C \setminus \{x\}| - vt(x, C) = r - 1 + q;\end{aligned}$$

$t(x, C)$ accounts for those triangles that $x$ closes with two other vertices in $C$. Since, $x$ is not connected to $v$, we have that $t(x, C) = t(x, C \cup \{v\})$. $t(x, V)$ accounts for the number of triangles that $x$ closes with the rest of vertices in $V$. These are $t(x, C)$ plus those triangles that vertex $x$ closes with another vertex of $C$ and a vertex of $V \setminus C$, plus those triangles that vertex $x$ closes with two other vertices in $V \setminus C$. $vt(x, V)$ accounts for the number of vertices in $V$ with whom $x$ closes at least one triangle, which are the neighbors of $x$ in $C$ and those $t$ vertices with whom $x$ is connected. Finally, we have that the union of vertices in $C \cup \{v\}$ and vertices in $V$ with whom $x$ closes at least one triangle is $r + q$, and the union of vertices in $C$ and vertices in $V$ with whom $x$ closes at least one triangle is $r + q - 1$. Therefore,

$$\begin{aligned}\Theta_2 = & \quad WCC(x, C \cup \{v\}) - WCC(x, C)\\ = & \quad \tfrac{t(x,C\cup\{v\})}{t(x,V)} \cdot \tfrac{vt(x,V)}{|C\cup\{v\}\setminus\{x\}|+vt(x,V\setminus\{C\cup\{v\}\})} - \\ & \quad \tfrac{t(x,C)}{t(x,V)} \cdot \tfrac{vt(x,V)}{|C\setminus\{x\}|+vt(x,V\setminus C)} = \\ = & \; -\tfrac{(r-1)(r-2)\delta^3}{(r-1)(r-2)\delta^3+q(q-1)\omega+q(r-1)\delta\omega} \cdot \tfrac{(r-1)\delta+q}{(r+q)(r-1+q)}.\end{aligned}$$

(iii) If $x = v$ we have

$$\begin{aligned}&t(x, C \cup \{v\}) = d_{in}(d_{in} - 1)\delta;\\ &t(x, V) = d_{in}(d_{in} - 1)\delta + d_{out}(d_{out} - 1)\omega + d_{out}d_{in}\omega;\\ &vt(x, V) = d_{in} + d_{out};\\ &vt(x, V) + |C \setminus \{x\}| - vt(x, C) = r + d_{out};\end{aligned}$$

In this case, $t(x, C\cup\{v\})$ accounts for those triangles that $x$ closes with $C$, with whom it is connected to $d_{in}$ vertices. $t(x, V)$ are those vertices vertex $x$ closes with $V$, which are those $x$ closes with $C$ plus those $x$ closes with other two vertices in $V \setminus C$. $vt(x, V)$ accounts for the number of vertices in $V$ with whom $x$ closes at least one triangle, which are $d_{in}$ plus $d_{out}$ since we assume that every edge closes at least one triangle. Finally, the union between the vertices in $C$ and those vertices in $V$ with whom $x$ closes at least one triangle is $r + d_{out}$. Therefore,

$$\begin{aligned}\Theta_3 = & \quad WCC(v, C \cup \{v\}) \\ = & \quad \tfrac{t(x,C\cup\{v\})}{t(x,V)} \cdot \tfrac{vt(x,V)}{|C|+vt(x,V\setminus C)} = \\ = & \quad \tfrac{d_{in}(d_{in}-1)\delta}{d_{in}(d_{in}-1)\delta+d_{out}(d_{out}-1)\omega+d_{out}d_{in}\omega} \cdot \tfrac{d_{in}+d_{out}}{r+d_{cout}}.\end{aligned}$$

□

## K. STATISTICAL INDICATORS

Given the following definitions:

— $t(S)$ is the number of triangles in set $S$.
— $N(x)$ is the set of neighbors of x.

we formally define the statistical indicators used in this paper.

**Triangle Density**:

$$\frac{3 \cdot t(S)}{|S| \cdot (|S|-1) \cdot (|S|-2)}$$

**Average Edge Density**:

$$\frac{1}{|S|} \cdot \sum_{x \in S} \frac{|N(x) \cap S|}{|S|-1}$$

**TPR**:

$$\frac{|\{x \in S : |t(x,S)| > 0\}|}{|S|}$$

**Expansion**:

$$\frac{\sum_{x \in S} |N(x) \cap (G \setminus S)|}{|S|}$$

**Conductance**:

$$\frac{\sum_{x \in S} |N(x) \cap (G \setminus S)|}{\sum_{x \in S} |N(x)|}$$

**Flake ODF**:

$$\frac{|\{x \in S : |N(x) \cap S| < N(x)/2\}|}{|S|}$$

**Average Inverse Edge Cut**:

$$\frac{1}{|S|} \cdot \sum_{x \in S} \frac{|N(x) \cap S|}{|N(x)|}$$

**Modularity per Community**:

$$\frac{1}{|S| \cdot (|S|-1)} \cdot \frac{1}{2 \cdot |E|} \cdot \left( \sum_{x \in S} |N(x) \cap S| - \sum_{x,y \in S} \frac{N(x) \cdot N(y)}{2 \cdot |E|} \right)$$

**Normalized Diameter**:

$$\frac{diameter(S)}{log(|S|) + 1}$$

**Bridge Ratio**:

$$\frac{2 \cdot bridges(S)}{\sum_{x \in S} |N(x) \cap S|}$$