Fully generalized graph cores

Alexandre P. Francisco and Arlindo L. Oliveira

INESC-ID / CSE Dept, IST, Tech Univ of Lisbon
Rua Alves Redol 9, 1000-029 Lisboa, PT
{aplf,aml}@inesc-id.pt

Abstract. A core in a graph is usually taken as a set of highly connected vertices. Although general, this definition is intuitive and useful for studying the structure of many real networks. Nevertheless, depending on the problem, different formulations of graph core may be required, leading us to the known concept of generalized core. In this paper we study and further extend the notion of generalized core. Given a graph, we propose a definition of graph core based on a subset of its subgraphs and on a subgraph property function. Our approach generalizes several notions of graph core proposed independently in the literature, introducing a general and theoretical sound framework for the study of fully generalized graph cores. Moreover, we discuss emerging applications of graph cores, such as improved graph clustering methods and complex network motif detection.

1 Introduction

The notion of k-core was proposed first by Seidman [1] for unweighted graphs in 1983. We say that a subgraph H of a graph G is a *k-core* or a *core of order k* if and only if H is a maximal subgraph such that $d(v) \geq k$, for all vertices v in H and where $d(v)$ is the degree of v with respect to H. Although general, this definition is intuitive and useful for the study of the structure of many real networks, with the first applications appearing in the area of social sciences [2]. More recently Batagelj and Zaveršnik [3] introduced the notion of generalized cores, allowing the definition of cores based on a vertex property function. In this paper, we further extend the concept of generalized core. Instead of vertex property functions, we consider subgraph property functions in general, leading to fully generalized cores. Moreover, we show that many notions of graph core and densely connected subgraphs proposed independently in the literature can be defined as fully generalized cores.

Given a graph $G = (V, E)$, possibly weighted, the problem consists of finding highly connected subgraphs. This problem is well known in graph clustering, where these subgraphs play the role of cluster or community cores. In particular, given a core for G, we can obtain a clustering for G by taking each core connected component as a seed set and by applying local partitioning methods [4, 5]. The notion of core has also been used in the context of multilevel schemata for graph clustering, where coarsening schemata were found to be closely related to the problem of core enumeration [6, 7]. The main idea behind most of the existing

notions is to merge the vertices that are more similar, namely in what concerns connectivity. Since we can define several vertex similarity scores and we can take different merging strategies, there are many possible definitions of core. The notion of fully generalized core proposed in this paper becomes particularly useful in this context.

Cliques and, in particular, the clique percolation method by Palla *et al.* [8] to detect densely connected, and possibly overlapping, clusters or communities on networks, are also related to graph cores. A clique is a complete graph and, if it has k vertices, then it is called a k-clique. The idea behind the clique percolation is that the edges within a cluster or community are likely to form cliques, *i.e.*, highly connected subgraphs. Conversely, the edges among clusters should not form cliques. In this context, we say also that two k-cliques are adjacent if they share $k-1$ vertices and, thus, a k-clique community is the largest connected subgraph obtained by the union of adjacent k-cliques. The method can be extended to weighted graphs either by considering a threshold on edge weights or a threshold on clique weights, defined for instance as the geometric mean of the weights of all edges [9]. Another maximal clique based approach was recently proposed by Shen *et al.* [10] to uncover both the overlapping and hierarchical community structures of networks. This method uses an agglomerative approach, merging pairs of maximal cliques that maximize a given similarity score. The main drawbacks of these methods are that detecting maximal cliques is an NP-hard problem, even though the authors found that the method is fast in practice for sparse networks, and that taking cliques as cluster building blocks may be an assumption too strong for many real networks. As pointed out by Saito *et al.* [11], methods based on the computation of graph cores or its extensions, *e.g.* k-dense clusters, can be better than methods based on the computation of maximal cliques. In particular, both k-cores and k-dense cores are less restrictive than k-cliques. Here, we analyse also k-cliques, k-clique percolations and k-dense clusters since these notions are particular cases of fully generalized cores.

The notion of fully generalized core introduced in this paper is also closely related with network motifs, allowing for composed network motifs. In fact, we can think of fully generalized cores as subgraphs formed by merging together highly connected motifs. The role of subgraph property functions is precisely to evaluate motif connectedness with respect to some criteria. Recent works in network analysis have made it clear that large complex natural networks reveal many local topological patterns, regarded as simple building blocks in networks, and named motifs. These characteristic patterns have been shown to occur much more frequently in many real networks than in randomized networks with the same degree sequence. For example, Milo *et al.* [12] discovered frequent characteristic local patterns on biological networks, *i.e.*, network motifs, observing that certain motifs are more common on biological networks than in other complex networks, revealing basic structural elements of such networks. Many efforts were done in order to understand the importance of network motifs [13, 14] and promising results were achieved, in spite of the rather limited network motifs that were used. For instance, Saito *et al.* [13] used only five predefined network motifs of size three and Albert *et al.* [14] used only four predefined small network

motifs. Note that many relevant processes in biological networks correspond to the mesoscale and, therefore, it will be interesting to study larger network motifs. Most of current network motif finding algorithms [12, 15] are enumeration based and limited to the extraction of smaller network motifs. The first reason is that the number of potential network motifs increases exponentially with the motif size [16]. A second one is that interesting motifs occur repeatedly in a given network but not in other networks, namely in randomized ones [12]. A third reason is that finding a given motif is closely related to the subgraph isomorphism problem. These reasons make the application of enumeration based algorithms unpractical when we consider mesoscale network motifs. Although different from motifs, we may want to study the occurrence of graphlets instead. Usually, graphlets must be induced subgraphs while motifs may be partial subgraphs. See for instance the recent work by Milenković *et al.* [17]. Graphlet frequency and degree distribution has been shown to provide good network signatures, becoming useful for the study of complex networks and for comparing them against proposed network models. As mentioned for motifs, the notion of fully generalized core is also useful to study graphlet composed cores.

2 Preliminaries

A *graph* or an *undirected graph* G is a pair (V, E) of sets such that $E \subseteq V \times V$ is a set of *unordered* pairs. The elements of V are the *vertices* and the elements of E are the *edges*. In this paper we assume that a graph does not have several edges between the same two vertices, *i.e.*, it does not have multiple edges, or edges that start and end at same vertex, *i.e.*, loops. When E is a set of *ordered* pairs we say that G is a *directed graph*. In this case the edge (u, v) is different from the edge (v, u) since they have different directions.

Given a graph $G = (V, E)$, the vertex set of G is denoted by $V(G)$ and the edge set of G is denoted by $E(G)$. Clearly $V(G) = V$ and $E(G) = E$. The number of vertices of G is its *order*, denoted either by $|V|$ or by $|G|$, and its number of edges is denoted by $|E|$. We say that a graph G is *sparse* if $|E| \ll |V|^2$. Two vertices $u, v \in V(G)$ are *adjacent* or *neighbors* if (u, v) is an edge, *i.e.*, $(u, v) \in E(G)$. Given a vertex $v \in V$, its set of neighbors is denoted by $N_G(v)$, or by $N(v)$ when G is clear from the context. The number of neighbors of v is its *degree* denoted by $d_G(v)$, or by $d(v)$ or d_v when G is clear from the context, *i.e.*, $d(v) = |N(v)|$. Given $V' \subseteq V(G)$, $d(V')$ denotes the sum of $d(v)$ for each $v \in V'$, *i.e.*, $d(V') = \sum_{v \in V'} d(v)$.

Let us now recall some graph properties. A graph G is *complete* or a *clique* if all vertices of G are pairwise adjacent. Usually, if G is complete and $|G| = n$, we denote G by K_n. Two graphs G and G' are *isomorphic*, denoted by $G \simeq G'$, if there is a bijection $\eta : V(G) \to V(G')$ such that $(u, v) \in E(G)$ if and only if $(\eta(u), \eta(v)) \in E(G')$, for all $u, v \in V$. Sometimes we are only interested in the notion of subgraph. G' is said to be a *subgraph* of G and G a *supergraph* of G' if $V(G') \subseteq V(G)$ and $E(G') \subseteq \{(u, v) \in E(G) \mid u, v \in V(G')\}$. G' is said to be a *proper subgraph* if $V(G') \subsetneq V(G)$. Given $V' \subseteq V(G)$, the subgraph *induced* by V' is the graph $G' = (V', E')$ where $E' = \{(u, v) \in E(G) \mid u, v \in V'\}$.

A *weighted graph* G is a tuple (V, E, w) where V and E form a graph $G = (V, E)$ and $w : E \to \mathbb{R}$ is a function that assigns to each edge $e \in E$ a weight $w(e)$. Note that we could also assign weights to the vertices or even arbitrary labels to both vertices and labels.

A *vertex similarity function* σ maps each pair of vertices to a positive real value, $\sigma : V^2 \longrightarrow \mathbb{R}_0^+$. Note that σ may be different from, although usually related to, the edge weight function w. Since σ reflects the similarity between two vertices $u, v \in V$, we usually say that u and v are the more similar the higher the value $\sigma(u, v)$. Moreover, we ignore pairs of vertices $u, v \in V$ for which $\sigma(u, v)$ is 0.0. The choice of the σ functions will always depend on the problem under study. For instance, we can simply use the vertex degree or the edge weights, if a suitable edge weight function w is provided. But, in general, these are not enough. For instance, we can consider a structural similarity function based on the cosine similarity. Note that we could start with other similarity functions, *e.g.*, with the Jaccard-Tanimoto index [18, 19]. Let w be the edge weight function. Given two connected vertices $(u, v) \in E$, their *structural similarity* $\sigma(u, v)$ is given by

$$\sigma(u, v) = \frac{2\, w(u, v) + \sum_{x \in N(u) \cap N(v)} w(u, x) w(v, x)}{\sqrt{1 + \sum_{x \in N(u)} w(u, x)^2} \sqrt{1 + \sum_{x \in N(v)} w(v, x)^2}}. \tag{1}$$

This equation reflects the cosine similarity between the neighborhoods of u and v. The term $2\, w(u, v)$ in the numerator and the 1's in the denominator were introduced to reflect the connection between u and v, being the only difference with respect to the usual definition of cosine similarity. In particular, if we extend this definition to all distinct pairs of vertices $u, v \in V$ or if we consider directed graphs, we may want to drop these terms. The version of Eq. (1) for unweighted graphs was first proposed by Xu *et al.* [20]. The similarity function σ as defined in Eq. (1) takes values in $[0, 1]$ and, given $(u, v) \in E$, $\sigma(u, v)$ grows as u and v share more neighbors. If u and v share all neighbors with equal weights, $\sigma(u, v)$ is 1.0. In particular, $\sigma(u, v)$ is 1.0 even if u and v share all neighbors through equal lowly weighted edges. In order to distinguish common neighbors connected through lowly weighted edges from common neighbors connected through highly weighted edges, we can compute the average weight among the common neighbors

$$\overline{w}(u, v) = \frac{w(u, v) + \sum_{x \in N(u) \cap N(v)} w(u, x) + w(v, x)}{1 + |N(u) \cap N(v)|} \tag{2}$$

and redefine σ as the product of $\overline{w}(u, v)$ by Eq. (1). Note that we may consider other terms instead of the weight average. For instance we could compute the maximum weight. Note also that σ as redefined above only takes values in $[0, 1]$ if w also takes values in $[0, 1]$.

We say that the subgraph H of G induced by $C \subseteq V$ is a *k-core* or a *core of order k* if and only if $d_H(v) \geq k$, for all $v \in C$, and H is a maximal subgraph with this property. The notion of k-core was proposed first by Seidman [1] for unweighted graphs. Usually, by abuse of nomenclature, each connected component of H is also called a k-core. More recently, Batagelj and Zaveršnik [3]

proposed a generalization of the notion of core, allowing the use of other properties of vertices than their degree. A *vertex property function* p on V is such that $p : V \times 2^V \longrightarrow \mathbb{R}$. Given $\tau \in \mathbb{R}$, a subgraph H of G induced by $C \subseteq V$ is a *τ-core with respect to p*, or a *p core at level τ*, if $p(v, C) \geq \tau$ for all $v \in C$ and H is a maximal subgraph with this property.

Given a vertex property function p, we say that p is *monotone* if and only if, given $C_1 \subseteq C_2 \subseteq V$, $p(v, C_1) \leq p(v, C_2)$ for all $v \in V$. Then, given a graph G, a monotone vertex property function p and $\tau \in \mathbb{R}$, we can compute the τ-core of G with respect to p by successively deleting vertices with p value lower than τ:

1. set $C \leftarrow V$;
2. while exists $v \in C$ such that $p(v, C) < \tau$, set $C \leftarrow C \setminus \{v\}$.

Theorem 1. *Given a graph G, a monotone vertex property function p and $\tau \in \mathbb{R}$, the above procedure determines the τ-core with respect to p.*

Corollary 1. *Given a monotonic vertex property function p and $\tau_1, \tau_2 \in \mathbb{R}$ such that $\tau_1 < \tau_2$, the cores are nested,* i.e.*, $C_2 \subseteq C_1$.*

These two results are due to Batagelj and Zaveršnik [3] and are particular cases of the more general results presented in this paper.

We can devise many vertex property functions. Here, we discuss three examples. Given a graph $G = (V, E)$, we can recover the classical definition of k-core by defining the vertex property function

$$p(v, C) = d_H(v), \tag{3}$$

where H is the subgraph of G induced by C. Thus, given $k \in \mathbb{N}$, a k-core with respect to p is precisely a classical k-core as defined by Seidman. Given a vertex similarity function σ, we can extend Eq. (3) as

$$p(v, C) = \sum_{u \in N(v) \cap C} \sigma(v, u). \tag{4}$$

Note that, taking σ as the weight function w, Eq. (4) is the natural extension of the k-core notion to weighted graphs leading to the notion of τ-core with $\tau \in \mathbb{R}$.

As in Eq. (1), the similarity function may already evaluate how strongly a vertex is connected to its neighbors. Thus, we may prefer the property function

$$p(v, C) = \max_{u \in N(v) \cap C} \sigma(v, u). \tag{5}$$

In this case, for $\tau \in \mathbb{R}$, all vertices v in the τ-core H are connected to some other vertex u in H such that $\sigma(u, v) \geq \tau$. With the vertex property function (5) the problem of finding cores becomes closely related to graphic matroids [21, 22]. In particular, taking σ as the weight function, a τ-core H is the maximal subgraph of G such that all edges in a maximum spanning forest of H have weight higher than τ. There are two efficient and well known approaches to enumerate the cores. We can sort the pairs of distinct vertices $u, v \in V$ by decreasing order of

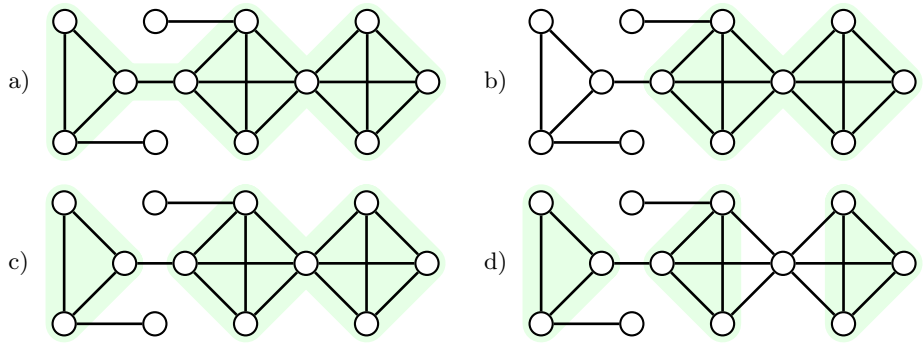

Fig. 1. Cores for different vertex property functions: a) 2-core with respect to the vertex property function (3); b) 3-core with respect to Eq. (3); d) 0.75-core with respect to Eq. (5); d) 0.85-core with respect to Eq. (5).

$\sigma(u, v)$ and iteratively merge them to form cores, which is the principle behind the algorithm of Kruskal [23]. Or we can iteratively visit each vertex u and merge it with the neighbor v that maximizes $\sigma(u, v)$, an approach related to the algorithm of Borůvka [24]. Thus, as is well known, both these approaches take $O(m \log n)$ where n is the number of vertices and m is the number of pairs such that $\sigma(u, v) > 0$. Note also that, if we consider the algorithm of Kruskal, we can get the full core hierarchy in a single run. We just need to store the cores for the thresholds we are interested in or, if preferred, the full dendrogram.

The examples of vertex property functions above are all monotonic, *i.e.*, $p(v, C_1) \leq p(v, C_2)$ for $C_1 \subseteq C_2 \subseteq V$. This is a straightforward consequence from the fact that, for the first two examples, σ is always positive and p is additive with respect to C and, in the third example, that the maximum can only increase as C grows.

3 Fully generalized cores

Let us now extend the notion of generalized core. Recently, Saito *et al.* [11] studied k-dense communities, where each pair of adjacent vertices must share at least $k-2$ neighbors. This is clearly related to the k-core notion. The difference is that they consider pairs of connected vertices instead of a single vertex. Moreover, Saito *et al.* pointed out that an extension would be the use of cliques, in general, instead of vertices or edges. Here, we further exploit these ideas and we propose an extension of generalized cores, allowing the evaluation of density for any subgraph. Let $G = (V, E)$ be a graph and let 2^G denote the set of subgraphs of G. Given $\mathcal{M} \subseteq 2^G$ a set of subgraphs of G, for instance a set of motifs, a *subgraph property function* p *over* $\mathcal{M}$ is such that $p : \mathcal{M} \times 2^G \longrightarrow \mathbb{R}$. We say that p is *monotone* if and only if the following conditions hold:

1. if H_1 is subgraph of $H_2 \in 2^G$, then $p(M, H_1) \leq p(M, H_2)$, for all $M \in \mathcal{M}$;
2. if $L_1 \in \mathcal{M}$ is subgraph of $L_2 \in \mathcal{M}$, then $p(L_1, H) \geq p(L_2, H)$, for all $H \in 2^G$.

The first condition is the generalization of the monotonicity condition discussed in the previous section. The second condition will allow us to refine cores with respect to p by changing the set of subgraphs $\mathcal{M}$, as stated in Proposition 1 and depicted in Fig. 2.

Let H be a subgraph of G, *i.e.*, $H \in 2^G$. We define $\mathcal{M}(H)$ as the set of subgraphs of H in $\mathcal{M}$, *i.e.*, $\mathcal{M}(H) = \mathcal{M} \cap 2^H$. Given $\tau \in \mathbb{R}$, H is a *τ-core with respect to p*, or a *p core at level τ*, if

1. $V(H) \subseteq \bigcup_{M \in \mathcal{M}(H)} V(M)$,
2. $p(M, H) \geq \tau$, for all $M \in \mathcal{M}(H)$,
3. and H is a maximal subgraph of G with properties 1 and 2.

The first condition states that H must be a subgraph of G induced by a set of subgraphs in $\mathcal{M}$. The second condition ensures that all subgraphs of H in $\mathcal{M}$ are densely connected within H and with respect to p. Finally, the third condition requires that H is maximal, *i.e.*, that there is not any τ-core H' with respect to p such that H is subgraph of H'. As before, by abuse of nomenclature, each connected component of H may also be called a core.

Given a graph G, $\mathcal{M} \subseteq 2^G$, a monotonic subgraph property function p over $\mathcal{M}$ and $\tau \in \mathbb{R}$, we can compute the τ-core H of G with respect to p as follows:

1. set H as the subgraph of G induced by $\bigcup_{M \in \mathcal{M}} V(M)$, *i.e.*, initialize H as the subgraph of G induced by the vertices of all subgraphs in $\mathcal{M}$;
2. while exists $M \in \mathcal{M}(H)$ such that $p(M, H) < \tau$, set H as the subgraph of G induced by $\bigcup_{M' \in \mathcal{M} \setminus \{M\}} V(M')$, *i.e.*, remove M from the list of subgraphs under consideration.

Theorem 2. *Given a graph G, $\mathcal{M} \subseteq 2^G$, a monotonic subgraph property function p over $\mathcal{M}$ and $\tau \in \mathbb{R}$, the above procedure determines the τ-core wrt p.*

Proof. Let H be the core returned by the procedure. We must show that

1. $p(M, H) \geq \tau$, for all $M \in \mathcal{M}(H)$;
2. H is maximal and independent of the order of deletions, *i.e.*, unique.

It is clear that 1 holds since all subgraphs M such that $p(M, H) < \tau$ are deleted in the procedure. Let us show that 2 also holds by absurd. Suppose that exists H' also determined by the above procedure, but such that $H' \neq H$. Thus, we have either $\mathcal{M}(H') \setminus \mathcal{M}(H) \neq \emptyset$ or $\mathcal{M}(H) \setminus \mathcal{M}(H') \neq \emptyset$. Let $M \in \mathcal{M}(H') \setminus \mathcal{M}(H)$ and $M_1, \ldots, M_k$ be the sequence of subgraphs removed by the procedure to obtain H. Since $M \in \mathcal{M}(H') \setminus \mathcal{M}(H)$, we have that $M \notin \mathcal{M}(H)$ and, thus, $M = M_j$ for some $1 \leq j \leq k$ (M is one of the removed subgraphs). Let $\mathcal{U}_0 = \emptyset$ and $\mathcal{U}_i = \mathcal{U}_{i-1} \cup \{M_i\}$, for $1 \leq i \leq k$. Note that $\mathcal{M}(G) \setminus \mathcal{U}_k = \mathcal{M}(H)$ and, given the deletion condition in the procedure, it is clear that $p(M_i, H_{i-1}) < \tau$, for $1 \leq i \leq k$, where H_{i-1} is the subgraph of G induced by the vertices of all subgraphs in $\mathcal{M}(G) \setminus \mathcal{U}_{i-1}$. Since $\mathcal{M}(H') \subseteq \mathcal{M}(G)$ and p is monotone, we also have that $\mathcal{M}(H') \setminus \mathcal{U}_{i-1} \subseteq \mathcal{M}(G) \setminus \mathcal{U}_{i-1}$ and $p(M_i, H'_{i-1}) < \tau$, for $1 \leq i \leq k$, where H'_{i-1} is the subgraph of H_{i-1} induced by the vertices of all subgraphs in $\mathcal{M}(H') \setminus \mathcal{U}_{i-1}$. In particular $p(M, H'_{j-1}) < \tau$ and, thus, M should be removed

in the procedure. Hence, if H' was returned, we have that $M \notin \mathcal{M}(H')$ for any $M \in \mathcal{M}(H') \setminus \mathcal{M}(H)$ – an absurd. So, $\mathcal{M}(H') \setminus \mathcal{M}(H) = \emptyset$ and, by an analogous argument, $\mathcal{M}(H) \setminus \mathcal{M}(H') = \emptyset$, *i.e.*, $\mathcal{M}(H) = \mathcal{M}(H')$ and $H = H'$. Therefore, H is unique, independent of the order of subgraph removal and maximal by construction, *i.e.*, 2 holds. ∎

Corollary 2. *Given a monotonic subgraph property function p and $\tau_1, \tau_2 \in \mathbb{R}$ such that $\tau_1 < \tau_2$, the τ_1-core H_1 and the τ_2-core H_2 with respect to p are nested,* i.e.*, H_2 is subgraph of H_1.*

Proof. By Theorem 2 we have that H_1 and H_2 are unique and independent of the order of deletions. Thus, since $\tau_1 < \tau_2$, we may apply the procedure to obtain H_1 and, by continuing the procedure, we may remove more subgraphs to obtain H_2. Therefore, H_2 is a subgraph of H_1. ∎

Although a subgraph property function p is only required to be defined over a set of subgraphs $\mathcal{M}$, the following result holds whenever p is extensible to any set $\mathcal{M}$, namely $p : 2^G \times 2^G \longrightarrow \mathbb{R}$ is well defined.

Proposition 1. *Let G be a graph, p be a monotonic subgraph property function over 2^G, $\tau \in \mathbb{R}$ and $\mathcal{M}, \mathcal{M}' \subseteq 2^G$. If all subgraphs $M' \in \mathcal{M}'$ can be induced by a sequence of subgraphs $M_1, \ldots, M_k \in \mathcal{M}$,* i.e.*, M' is a subgraph induced by $\bigcup_{i=1}^{k} V(M_i)$, then the τ-core H' with respect to p over $\mathcal{M}'$ is a subgraph of the τ-core H with respect to p over $\mathcal{M}$.*

Proof. Since H' is a τ-core with respect to p over $\mathcal{M}'$, there are $M'_1, \ldots, M'_\ell \in \mathcal{M}'$ such that H' is the subgraph induced by $\bigcup_{j=1}^{\ell} V(M'_j)$ and $p(M'_j, H') \geq \tau$, for $1 \leq i \leq \ell$. By hypothesis, each M'_j is a subgraph induced by $\bigcup_{i=1}^{k} V(M_i)$, where $M_1, \ldots, M_k \in \mathcal{M}$. Then, since p is monotone, $p(M_i, H') \geq p(M'_j, H') \geq \tau$, for $1 \leq i \leq k$, and thus $M_1, \ldots, M_k$ are part of the τ-core with respect to p over $\mathcal{M}$. Therefore, all $M_1, \ldots, M_k$, for all M'_j with $1 \leq j \leq \ell$, are subgraphs of H, *i.e.*, H' is subgraph of H. ∎

By Proposition 1, given a suitable subgraph property function, we are able to incrementally build the τ-core by refining the set of subgraphs $\mathcal{M}$. For instance, let p be the subgraph property function

$$p(M, H) = |V(M) \cap V(H)| + |X \cap V(H)|, \tag{6}$$

where $X = \bigcap_{u \in V(M)} N_G(u)$. Note that p is monotone only if we restrict M to cliques. Taking $\mathcal{M}$ as the set of singleton subgraphs, *i.e.*, $\mathcal{M} = \{(\{u\}, \emptyset) \mid u \in V\}$, Eq. (6) is equivalent to Eq. (3) minus one. Thus, given $k \in \mathbb{N}$, a k-core with respect to p over $\mathcal{M}$ is a classical $(k-1)$-core as defined by Seidman. If we take $\mathcal{M}'$ as the set of subgraphs induced by E, *i.e.*, $\mathcal{M}' = \{(\{u, v\}, \{(u, v)\}) \mid (u, v) \in E\}$, the k-cores with respect to p over $\mathcal{M}'$ are precisely the k-dense communities as proposed by Saito *et al.*. Note that, given $k \in \mathbb{N}$, the k-dense community requires that two connected vertices share at least $k-2$ neighbors. In a way analogous to the method proposed by Saito *et al.*, we can compute a k-core with respect to p

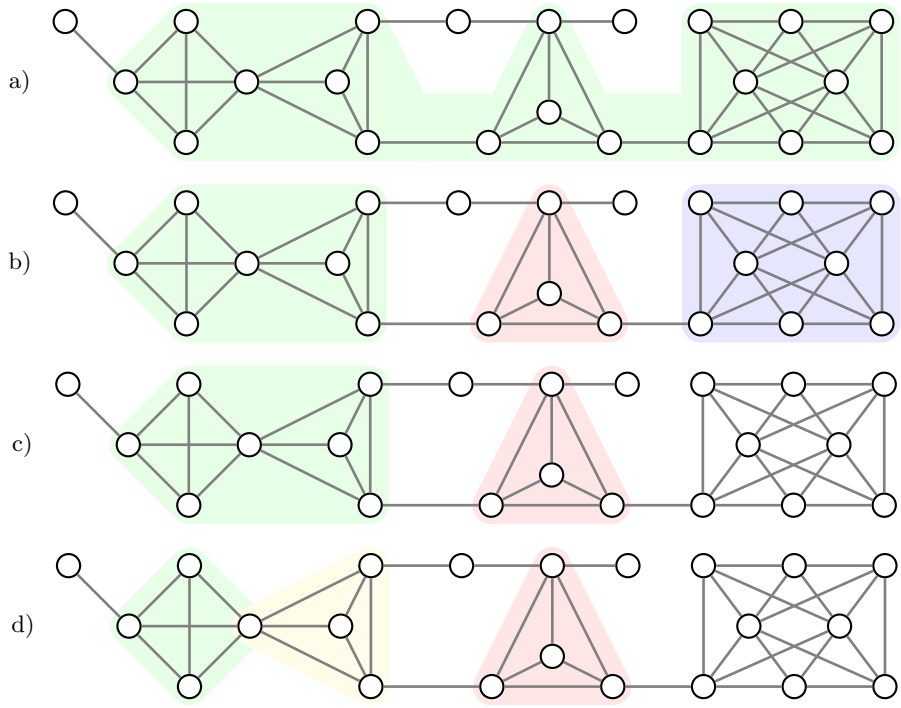

Fig. 2. Graph cores identified using the subgraph property function (6) and different sets of subgraphs $\mathcal{M}$, including the comparison with k-clique percolations. The shadowed vertices are: a) a 4-core with respect to (6) over $\mathcal{K}_1$, *i.e.*, a classical 3-core; b) a 4-core with respect to (6) over $\mathcal{K}_2$, *i.e.*, three classical 4-dense communities; c) a 4-core with respect to (6) over $\mathcal{K}_3$; d) three 4-clique percolation communities. Note that the clique percolations in d) are subgraphs of the core in c), which is subgraph of the core in b), which is also subgraph of the core in a).

over $\mathcal{M}$ and, then, we can refine it to obtain a k-dense community by computing a k-core with respect to p over $\mathcal{M}'$. This is a straightforward application of Proposition 1, as illustrated by the two first cases in Fig. 2.

Clearly, we can consider any set of subgraphs with the subgraph property function (6). For instance, given $\ell \in \mathbb{N}$, let $\mathcal{K}_\ell$ be the set of subgraphs of G isomorphic to the clique $K_{\ell'}$ of size ℓ', for all $\ell' \leq \ell$, *i.e.*,

$$\mathcal{K}_\ell(G) = \{H \mid H \text{ is subgraph of } G,\ H \simeq K_{\ell'} \text{ and } \ell' \leq \ell\}. \tag{7}$$

Note that if we consider $\ell = 1$ or $\ell = 2$, we recover the definitions of classical k-core and k-dense community, respectively. Moreover, for any $k \in \mathbb{N}$, each vertex in the k-core with respect to p over $\mathcal{K}_{k-1}$ belongs to at least one k-clique. This is interesting since it is closely related to the communities found with the clique percolation method [8]. In particular a k-clique percolation community is a subgraph of the k-core with respect to p over $\mathcal{K}_{k-1}$ and, by Proposition 1, it is a subgraph of the k-core with respect to p over $\mathcal{K}_\ell$ for any $\ell < k$ (see Fig. 2). In

particular, this establishes a relation of nesting between classical k-cores, k-dense communities and k-clique communities.

As we did for Eq. (3), we can easily extend Eq. (6) to weighted graphs. Given a vertex similarity function σ, the subgraph property function becomes

$$p(M, H) = \sum_{u \in V(M)} \sum_{v \in X \cap V(H)} \sigma(u, v), \tag{8}$$

where $X = \bigcap_{u \in V(M)} N_G(u)$. Note that p is monotone only if the weights are equal to 1.0, otherwise the second monotonocity contidion may not hold. Taking σ as the weight function w and considering $\mathcal{M} = \mathcal{K}_2$, Eq. (8) is the natural extension of the k-dense notion to weighted graphs leading to the notion of τ-dense community with $\tau \in \mathbb{R}$.

The Corollary 2 (or 1 in the simpler case) ensures that, given a monotonic subgraph property function p, we can built a hierarchy of nested cores by considering different values of τ. This is interesting since, by ranging over different values of τ, we get a hierarchy of cores.

4 Discussion and applications

In this paper we propose fully generalized cores, which extend several core definitions proposed in the literature under a common framework. Moreover, we discuss a greedy approach to solve the problem of identifying fully generalized cores. The complexity of this approach is clearly dependent on subgraph property functions, which may be computationally costly. Although for some subgraph property functions this problem can be stated as graphic matroid [21, 22], it remains to be seen under which formal conditions this combinatorial problem becomes a matroid.

In what concerns interesting and desirable properties, there are other related approaches to core enumeration. Recently Xu *et al.* [20] implicitly proposed the following alternative definition of core. Given the similarity function (1), $n \in \mathbb{N}$ and $\varepsilon > 0$, we say that $(u, v) \in E$ is a core edge if $\sigma(u, v) \geq \varepsilon$, and that $u \in V$ is a core vertex if $|\{v \in N(u) \mid \sigma(u, v) \geq \varepsilon\}| \geq n$. Then, a set of vertices $C \subseteq V$ is a core in G if all $u \in C$ is a core vertex and if, for all $u, v \in C$, there is a connecting path composed only of core edges. The parameter n is the main difference with respect to the core enumeration approaches discussed in this paper. Given $n \in \mathbb{N}$, we compute the ε-core H with respect to the property function (5), but we further filter it by leaving just the vertices $u \in V$ such that $|\{v \in H \mid \sigma(u, v) \geq \varepsilon\}| \geq n$. Thus, although the definition of core proposed by Xu *et al.* is related to the notion of generalized core, it introduces an extra degree of freedom that is interesting if we require higher resolutions.

There are several interesting applications for fully generalized cores. Here, we briefly discuss two of them. As discussed before, an application is the detection of densely connected subgraphs within graph clustering methods. Given a core, we can take each connected component as a seed set and apply well known local partition methods [25–27]. Note that by using the approach described in this

paper, we can get a hierarchy of cores and, thus, we are able to get a hierarchical clustering. There are several alternatives for hierarchical clustering and local optimization. For instance, Lancichinetti *et al.* [28] proposed a multiresolution method that optimizes a local fitness score by adding and removing vertices to increase the fitness score, following an approach like the one proposed by Blondel *et al.* [29]. These are equivalent to the approaches based on ranking, where each vertex constitutes a core or seed set. The main issue with these simpler approaches is that there is not any guarantee about their effectiveness. On the other hand, local ranking based on, *e.g.*, the heat kernel has supporting results both with respect to local optimization complexity and clustering quality [27]. These approaches allow also for the detection of vertices that appear in multiple clusters, *i.e.*, overlapping clusterings. Note also the ability to obtain local clusterings, in particular when we do not know all the graph. This problem is partially addressed by the local optimization or local clustering techniques. But an important issue remains: what happens if the seed set is composed by vertices already within an overlap? If we just use a standard local clustering approach, we will obtain just a big cluster composed of several smaller and overlapping clusters. By partially exploring the neighborhood of the seed set, by enumerating the cores, and by applying local clustering to the obtained seed sets, we can detect the smaller and overlapping clusters.

A second application is the detection of complex network motifs, which we already mentioned. Given a set of motifs or graphlets, we can enumerate the cores composed only by vertices belonging to these motifs or graphlets. The main task becomes defining a suitable subgraph property function. The resulting cores can then be statically evaluated, identifying possible mesoscale network motifs. This is of high importance since enumerating and evaluating motifs or graphlets with a reasonable size is computationally demanding. Unlike graphlets, network motifs may not be induced subgraphs and, thus, we may want to consider the merging of motifs instead of vertex induced subgraphs in our definition of fully generalized cores. The results presented herein remain valid.

References

1. Seidman, S.B.: Network structure and minimum degree. Social Networks **5**(3) (1983) 269–287
2. Wasserman, S., Faust, K.: Social network analysis: Methods and applications. Cambridge University Press (1994)
3. Batagelj, V., Zaveršnik, M.: Generalized cores. `arXiv:cs/0202039` (2002)
4. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: Natural cluster sizes and the absence of large well-define clusters. `arXiv:0810.1355` (2008)
5. Wei, F., Qian, W., Wang, C., Zhou, A.: Detecting Overlapping Community Structures in Networks. World Wide Web **12**(2) (2009) 235–261
6. Schloegel, K., Karypis, G., Kumar, V.: Graph partitioning for high-performance scientific simulations. Morgan Kaufmann Publishers, Inc. (2003)
7. Abou-Rjeili, A., Karypis, G.: Multilevel algorithms for partitioning power-law graphs. In: IEEE International Parallel & Distributed Processing Symposium, IEEE (2006) 10

8. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435** (2005) 814–818
9. Farkas, I., Ábel, D., Palla, G., Vicsek, T.: Weighted network modules. New J. Physics **9**(6) (2007) 180
10. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. Physica A: Statistical Mechanics and its Applications **388**(8) (2009) 1706–1712
11. Saito, K., Yamada, T., Kazama, K.: Extracting Communities from Complex Networks by the k-dense Method. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **91** (2008) 3304–3311
12. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network Motifs: Simple Building Blocks of Ccomplex Networks. Science **298** (2002) 824–827
13. Saito, R., Suzuki, H., Hayashizaki, Y.: Construction of reliable protein-protein interaction networks with a new interaction generality measure. Bioinformatics **19**(6) (2002) 756–763
14. Albert, I., Albert, R.: Conserved network motifs allow protein-protein interaction prediction. Bioinformatics **20**(18) (2004) 3346–3352
15. Kashtan, N., Itzkovitz, S., Milo, R., Alon, U.: Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. Bioinformatics **20**(11) (2004) 1746–1758
16. Kuramochi, M., Karypis, G.: An efficient algorithm for discovering frequent subgraphs. IEEE Transactions on Knowledge and Data Engineering **16**(9) (2004) 1038–1051
17. Milenković, T., Lai, J., Pržulj, N.: Graphcrunch: a tool for large network analyses. BMC Bioinformatics **9**(1) (2008) 70
18. Jaccard, P.: Distribution de la flore alpine dans le Bassin des Dranses et dans quelques regions voisines. Bull. Soc. Vaud. Sci. Nat **37** (1901) 241–272
19. Tanimoto, T.T.: IBM Internal Report 17th Nov. Technical report, IBM (1957)
20. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: Scan: a structural clustering algorithm for networks. In: SIGKDD, ACM (2007) 824–833
21. Whitney, H.: On the abstract properties of linear dependence. American Journal of Mathematics **57**(3) (1935) 509–533
22. Tutte, W.T.: Lectures on matroids. J. Res. Nat. Bur. Stand. B **69** (1965) 1–47
23. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the AMS **7**(1) (1956) 48–50
24. Borůvka, O.: On a minimal problem. Prace Moraské Pridovedecké Spolecnosti **3** (1926)
25. Spielman, D.A., Teng, S.H.: A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning. `arXiv.org:0809.3232` (2008)
26. Andersen, R., Lang, K.J.: Communities from seed sets. In: WWW, ACM (2006) 223–232
27. Chung, F.: The heat kernel as the pagerank of a graph. PNAS **104**(50) (2007) 19735–19740
28. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New J. Physics **11** (2009) 033015
29. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment (2008) P10008