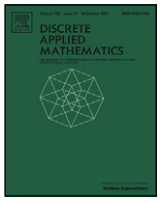
Discrete Applied Mathematics 160 (2012) 2389–2396

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam



Dense and sparse graph partition

Julien Darlay [a,b,*], Nadia Brauner [b], Julien Moncel [c,d,e]

[a] Bouygues e-lab, 32 avenue Hoche 75008 Paris, France

[b] Grenoble-INP / UJF-Grenoble 1 / CNRS, G-SCOP UMR5272, F-38031 Grenoble, France

[c] Fédération de recherche "maths à modeler", France

[d] CNRS, LAAS, 7 avenue du colonel Roche, F-31077 Toulouse Cedex 4, France

[e] Université de Toulouse, UPS, INSA, INP, ISAE, UT1, UTM, LAAS, F-31077 Toulouse Cedex 4, France

ARTICLE INFO

Article history:
Received 11 May 2011
Received in revised form 5 June 2012
Accepted 9 June 2012
Available online 4 July 2012

Keywords:
Graph partition
Community detection
Complexity
Algorithms

ABSTRACT

In a graph $G = (V, E)$, the density is the ratio between the number of edges $|E|$ and the number of vertices $|V|$. This criterion may be used to find communities in a graph: groups of highly connected vertices. We propose an optimization problem based on this criterion; the idea is to find the vertex partition that maximizes the sum of the densities of each class. We prove that this problem is NP-hard by giving a reduction from graph-k-colorability. Additionally, we give a polynomial time algorithm for the special case of trees.

1. Introduction

Let $G = (V, E)$ be a simple connected undirected graph; the density [16] of G is given by $d(G) = \frac{|E|}{|V|}$. Let $X \subseteq V$ and $E(X) = \{\{u, v\} \in E,\ u \in X,\ v \in X\}$. The subgraph induced by X is $G[X] = (X, E(X))$ and the complement graph $\overline{G}$ of the graph G is the graph on V with the edge set $\overline{E} = V \times V \setminus E$. Let Π be the set of all the partitions of V with no empty class. The *density* of a partition $P \in \Pi$ is given by:

$$d(P) = \sum_{X \in P} d(G[X]) = \sum_{X \in P} \frac{|E(X)|}{|X|}.$$

One can note that the definition of a graph density used here is different from another definition of density called edge density in [11]. The edge density of a graph with n vertices and m edges is defined as the ratio $\frac{2m}{n(n-1)}$. Our definition of density is also known as the ratio association criterion in the image segmentation literature [10,26]. In the literature of graph partition, the notion of sparse dense partition also exists as a particular graph bipartition [12].

The density can be used as a fitness function in the area of community detection. Many empirical problems can be modeled as networks that divide naturally into communities, for instance protein interactions, social interactions, etc [7,17,25]. Intuitively, a community is a set of nodes that are highly connected and only have a few links with nodes from the outside. Finding such groups provides help in understanding and visualizing the structure of the network.

The general problem of community detection is widely studied and various fitness functions have been proposed [4,21,22]. The large number of fitness functions can be explained by the impossibility theorem of Kleinberg [18]. It states that

* Corresponding author at: Grenoble-INP / UJF-Grenoble 1 / CNRS, G-SCOP UMR5272, F-38031 Grenoble, France. Fax: +33 4 76 57 46 95.
E-mail addresses: jdarlay@bouygues.com (J. Darlay), nadia.brauner@g-scop.grenoble-inp.fr (N. Brauner), julien.moncel@iut-rodez.fr (J. Moncel).

0166-218X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2012.06.004

it is not possible to find a clustering function for the partition of a graph that verifies properties of scale-invariance, richness and consistency. From a complexity point of view, most of the graph partition problems associated to community detection are NP-hard and several heuristics have been proposed [14,23,27]. When the graph is restricted to be a tree, polynomial algorithms can be devised for some fitness functions [20].

We faced this problem of community detection while analyzing data for clinical research in medicine [8]. We used the logical analysis of data method [3] where a large set of patterns is generated, a pattern being the characteristics of patients having similar properties for the studied pathology. Our objective was to group patterns representing almost the same sets of patients in order to decrease the size of the problem. This is modeled as community detection in a graph where each vertex is a pattern and an edge connects two vertices corresponding to similar patterns. In this framework, the notion of density which is defined above, is relevant in the sense that it aims at maximizing the density within the clusters rather than minimizing the inter-cluster density, which is the case for many notions of density in the literature.

The *sparsity* of a partition $P \in \Pi$ with $|P|$ classes is given by:

$$F(P) = \frac{|P|}{2} + d(P).$$

Notice that maximizing the density of a partition P in a graph G is equivalent to minimizing its sparsity in the complement graph $\overline{G}$. The proof is presented in the next section.

The usual optimization problem associated with the notion of density is to find a subgraph of maximum density. When the number of vertices in the subgraph is part of the input, the problem is NP-hard [13]. When the number of vertices in the subgraph is free, the problem can be solved in polynomial time using flow techniques [16] or linear programming [6]. These results motivate the use of the density as a partition objective function. From a practical point of view, the polynomial algorithms for the densest subgraph could be used to devise efficient heuristics for community detection.

The sparsity is a less classical objective function. When the number of classes is fixed $|P| = k$, the problem of minimizing F is equivalent to the minimization of the famous k-means criterion (see for instance [5,28]). In this case, the number of edges in a class is replaced by the sum of their weights in the definition of the density. This problem is shown to be NP-hard when the edges are weighted by the Euclidean distance [1,9].

In this paper we address the problem of finding a partition of maximum density, without fixing the number of classes of the partition. Indeed, when the number of classes is given, the problem is a generalization of a partition into k cliques [15]. We show that finding a partition of maximum density in a graph is equivalent to finding a partition of minimum sparsity in its complementary graph. We also derive some results about the NP-hardness (Theorem 1) and the non-approximability (Theorem 2) of these problems. We finally give a polynomial time algorithm for finding a partition of maximum density on a tree (Theorem 3).

2. Preliminaries

For the sake of clarity we recall some notions of graph theory and matching theory. All the definitions and theorems can be found in [11]. A *path* is a non-empty graph $P = (V, E)$ of the form $V = \{x_0, x_1, \ldots, x_n\}$ and $E = \{x_0x_1, x_1x_2, \ldots x_{n-1}x_n\}$. Let us denote by P_n a path containing n edges. A *star* is a tree where at most one vertex has degree greater than 1. A *matching* in an undirected simple graph is a set of independent edges. A vertex is *covered* by a matching if it is incident to one edge of the matching. A matching M is called a *perfect matching* if all vertices are covered by M. A *vertex cover* in G is a set $S \subseteq V$ such that each edge of G is incident to at least one vertex in S. A path $P = (V, E)$ in a graph G is an *alternating path* with respect to a matching M if $E \setminus M$ is a matching. An alternating path is an *augmenting path* if its endpoints are not covered by M. We give two classical theorems in matching theory (proofs omitted).

Theorem (*König [19]*)**.** *Let G be a bipartite graph. Then the maximum cardinality of a matching in G is equal to the minimum cardinality of a vertex cover.*

Theorem (*Petersen [24]*)**.** *Let G be a graph with a matching M. Then M is maximum if and only if there is no augmenting path.*

Let us define formally the decision problems we consider in the rest of the paper.

Dense Graph Partition

Instance. An undirected graph $G = (V, E)$ and a positive rational D.

Question. Is there a partition $P \in \Pi$ such that $d(P) \geq D$?

Sparse Graph Partition

Instance. An undirected graph $G = (V, E)$ and a positive rational D.

Question. Is there a partition $P \in \Pi$ such that $F(P) \leq D$?

Graph-k-Colorability [15]

Instance. An undirected graph $G = (V, E)$.

Question. Is there a partition $P \in \Pi$ such that $|P| = k$ and for all $X \in P$, $G[X]$ is a stable set?

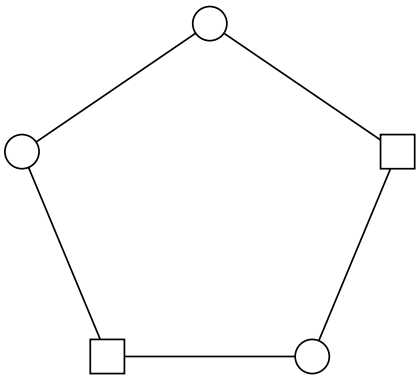


Fig. 1. A partition of the cycle on 5 vertices (C_5) with sparsity less than $\frac{3}{2}$. The color classes are represented by circles and squares.

We also consider the optimization versions of these problems which will be prefixed by Min or Max. For instance:

Min Sparse Graph Partition

Instance. An undirected graph $G = (V, E)$.

Solution. A partition $P^* \in \Pi$ such that $F(P^*) \leq F(P),\ \forall P \in \Pi$.

A first observation shows that maximizing the density is equivalent to minimizing the sparsity using a simple transformation on the instance.

Property 1. *The optimization problem* Max Dense Graph Partition *of a graph G is equivalent to* Min Sparse Graph Partition *of $\overline{G}$ the complement graph of G.*

Proof. Let $\overline{E}(X)$ be the set of edges of the complement graph induced by the set of vertices X. One can rewrite the density of P in G using the set of edges of $\overline{G}$

$$d(P) = \sum_{X\in P} \frac{|E(X)|}{|X|} = \sum_{X\in P} \frac{\frac{|X|(|X|-1)}{2} - |\overline{E}(X)|}{|X|} = \sum_{X\in P} \left(\frac{|X|-1}{2} - \frac{|\overline{E}(X)|}{|X|} \right) \tag{1}$$

$$= \frac{n}{2} - \frac{|P|}{2} - \sum_{X\in P} d(\overline{G}[X]). \tag{2}$$

Thus $d_G(P) = \frac{n}{2} - F_{\overline{G}}(P)$ with $F_{\overline{G}}(P)$ being the sparsity of the complement graph G. □

Since the two problems are equivalent we focus on minimizing the sparsity of a graph G. Every coloring of G is a feasible solution and thus we obtain the following upper bound.

Property 2. *Let G be a graph and P^* a vertex partition of G that minimizes F. The following inequality holds:*

$$F(P^*) \leq \frac{\chi(G)}{2}.$$

Proof. Let P be the partition associated with a $\chi(G)$-coloring of G where each color is a class of P. Since each class of P is a stable set, its density is equal to 0 and $F(P) = \frac{\chi(G)}{2}$. Since $F(P^*) \leq F(P)$ the inequality holds. □

Notice that the bound is not always tight. For instance in the cycle on 5 vertices the optimal coloring uses 3 classes but there exists a partition of V of cardinality 2 with $F(P) = 1 + \frac{1}{3} < \frac{3}{2}$ (see Fig. 1).

3. NP-hardness and non-approximability

We show that the problem Sparse Graph Partition is NP-complete by giving a reduction from Graph-k-colorability. The decision problem Graph-k-colorability is known to be NP-complete (see for instance [15]). Since Sparse Graph Partition and Dense Graph Partition are equivalent by Property 1, it implies that Dense Graph Partition is also NP-complete. We first describe a graph transformation and a useful property for the reduction.

Let G be a simple undirected graph. We define G^q the graph constructed from G where each vertex v is replaced by a stable set of cardinality q: $\{v^1\, v^2\, \ldots\, v^q\}$. Each edge (i, j) of G is replaced by the complete bipartite subgraph: $(\{i^1\, \ldots\, i^q\},\ \{j^1\, \ldots j^q\})$; for instance the graph C_5 of Fig. 1 is transformed into the graph C_5^2 in Fig. 2. This transformation intends to increase the density without changing the chromatic number.

Lemma 1. *Let G be a graph and G^q the graph obtained from the transformation of G. Let $\chi(G)$ and $\chi(G^q)$ be their respective chromatic numbers. Then $\chi(G) = \chi(G^q)$.*

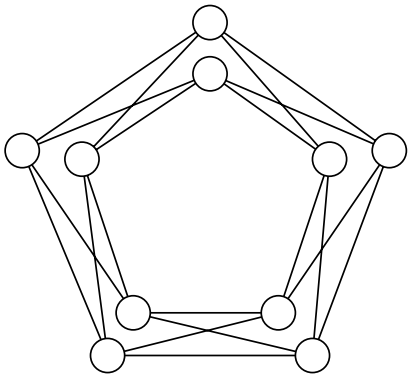


Fig. 2. The graph C_5^2.

Proof. The inequality $\chi(G^q) \leq \chi(G)$ is trivial since any coloring of G gives a coloring for G^q. Suppose now that $\chi(G^q) < \chi(G)$ then by keeping one vertex in each $(v^1 \dots v^q)$ we obtain a coloring of G with less than $\chi(G)$ colors, which is a contradiction. Thus $\chi(G^q) \geq \chi(G)$ and hence $\chi(G) = \chi(G^q)$. □

Now we show a reduction from Graph-k-colorability to Sparse Graph Partition that uses the previous transformation.

Theorem 1. *The* Sparse Graph Partition *problem is NP-complete.*

Proof. It is easy to see that Sparse Graph Partition is in NP since there exists a non-deterministic algorithm that can guess a partition $P \in \Pi$ and verify that $F(P) \leq D$ in polynomial time.

Let us consider an instance G of Graph-k-colorability. If G has less than k vertices then we are done. Otherwise we transform G into an instance of Sparse Graph Partition in polynomial time as follows.

Given a graph G on n vertices, we build the graph G^q with $q = n^4$. We claim that there exists a k-coloring of G if and only if there exists a partition P of G^q such that $F(P) \leq \frac{k}{2}$.

From a k-coloring of G one can derive a k-coloring of G^q. Hence from Property 2 there exists a partition P of the vertices of G^q with $F(P) \leq \frac{k}{2}$.

Conversely, suppose that we have a partition P of G^q such that $F(P) \leq \frac{k}{2} \leq \frac{n}{2}$, this implies $|P| \leq k$. Consider an edge (i, j) of G and the sets of vertices $I = \{i^1, \dots, i^q\}$ and $J = \{j^1, \dots, j^q\}$ of G^q. By the pigeonhole principle there exists a class C_I of P containing more than $\frac{q}{k} \geq \frac{q}{n} = n^3$ vertices from I. Let S_I be the set of vertices of I in C_I. Using the same argument, there exists a class C_J containing more than n^3 vertices from J. If $C_I = C_J$ then $d(C_I) = \frac{|E(C_I)|}{|C_I|} \geq \frac{|E(C_I)|}{|V(G^q)|} \geq \frac{n^3 \cdot n^3}{nq} \geq n \geq k > \frac{k}{2}$ and $F(P) \geq d(C_I) > \frac{k}{2}$ which is a contradiction. Thus for each edge (i, j) of G, the sets S_I and S_J belong to different classes of P. One can construct a proper coloring with k colors of G using the partition P of G^q and the set S_U for each vertex u.

Finally notice that G^q can be constructed from G in polynomial time. The sets S_U can be obtained from a partition P of G^q in polynomial time. Thus the reduction from Graph-k-colorability to Sparse Graph Partition is polynomial. □

Using slight modifications on the previous proof, we have the following theorem on the approximability of Min Sparse Graph Partition.

Theorem 2. *There is no polynomial-time r-approximation algorithm to* Min Sparse Graph Partition *problem for some constant r unless $P = NP$.*

Proof. Assume that a polynomial time algorithm can find a partition P such that $F(P) \leq rF(P^*)$. Using the same proof as Theorem 1 with $q = rn^4$ one can obtain an r-approximation of $\chi(G)$ for every graph G using the r-approximation algorithm on G^q. Consider an edge (i, j) of G and assume that the classes C_I and C_J are the same in P. Then $F(P) \geq d(C_I) \geq rn > r\frac{\chi(G^q)}{2} \geq rF(P^*)$ which is a contradiction. Then S_I and S_J belong to different classes of P and one can construct a proper coloring of G using $|P| \leq r\chi(G)$ colors. This reduction preserves the approximation and since Min Graph Coloring does not belong to *APX* [2], Min Sparse Graph Partition is not in *APX* either. □

Unfortunately, this proof could not be directly extended to the problem Max Dense Graph Partition since the reduction between the two problems does not keep approximability. As shown in Eq. (2) in the proof of Property 1, a constant appears in the relation between the density of partition P of a graph G and the sparsity of the same partition in the complement graph $\overline{G}$.

4. The polynomial case of trees

When dealing with NP-hard optimization problems, it is natural to take a look at special cases. In this section we give a polynomial time algorithm to find the partition with the maximum density when the graph is a tree. Partitions with maximum density are called *optimal partitions*. In the case of trees, these partitions have a strong link with well known

theoretical concepts such as matching and vertex cover. We first derive some properties about these optimal partitions, then we give a polynomial algorithm based on dynamic programming.

The following property gives lower and upper bounds on the density of a tree. These simple bounds are used to characterize the classes of an optimal partition (Lemma 4).

Property 3. *Let T be a tree such that $|V| > 1$ and $d(T)$ its density then we have:*

$$d(T) = \frac{m}{n} = \frac{n-1}{n} = 1 - \frac{1}{n}.$$

From this equality we can derive an upper bound on the density of any tree T: $d(T) < 1$ and a lower bound $d(T) \geq \frac{1}{2}$.

The following lemmas describe the structure of an optimal partition. They show that each class of P^* is a star.

Lemma 2. *Let G be a connected graph and let P^* be an optimal partition of G. Then for any class X of P^*, the graph $G[X]$ is connected.*

Proof. If $G[X]$ is not connected in a partition P, one can construct a new partition P' by replacing X by a new class for each connected component $X_1, \ldots, X_k$ of $G[X]$. The new partition is better than P:

$$d(P') - d(P) = \left(\sum_{i=1}^{k} \frac{|E(X_i)|}{|X_i|}\right) - \left(\frac{\sum_{i=1}^{k} |E(X_i)|}{\sum_{i=1}^{k} |X_i|}\right) \geq 0.$$

Since $\forall a_i \geq 0,\ b_i > 0$ we have $\sum_i (\frac{a_i}{b_i}) \geq \frac{\sum_i a_i}{\sum_i b_i}$. □

Lemma 3. *Let T be a tree and P^* an optimal partition of T. Then no class of P^* contains only an isolated vertex.*

Proof. Suppose there exists $C \in P^*$ such that $C = \{u\}$ with $u \in V$. Since T is connected, there exists a vertex v such that $\{u, v\}$ is an edge. Suppose that $v \in C'$ and let $C'' = C' \cup C$. Since C' is connected C'' is also connected. Then one has $d(C'') - d(C') - d(C) = \frac{|C'|}{|C'|+1} - \frac{|C'|-1}{|C'|} > 0$ and P^* is not optimal. □

Lemma 4. *Let T be a tree and P^* an optimal partition of T. Then for any class X of P^*, the graph $G[X]$ does not contain a path of length 3.*

Proof. Suppose there exists $X \in P^*$ such that $\{u, r, s, t\} \subseteq X$ and (u, r, s, t) is a P_3. By removing the edge (r, s) we create two connected components X' and X''. Since $G[X]$, $G[X']$ and $G[X'']$ are trees we have the following inequalities $d(X) < 1$ and $d(X') + d(X'') \geq \frac{1}{2} + \frac{1}{2}$ and P^* is not optimal. □

As a corollary, we get.

Corollary 1. *Let T be a tree and P^* an optimal partition of T. Then for each X, a class of P, $G[X]$ is a star.*

The following properties give upper bounds on the density of a bipartite graph and on an optimal partition.

Property 4. *Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. Then $d(G) \leq \frac{n}{4}$.*

Proof. We derive an upper bound on the density in the case of bipartite graphs:

$$d(G) = \frac{m}{n_1 + n_2} \leq \frac{n_1 n_2}{n_1 + n_2} \leq \frac{n_1 + n_2}{4}.$$

The first inequality is trivial and the second one comes from:

$$(n_1 + n_2)^2 - 4n_1 n_2 = (n_1 - n_2)^2 \geq 0. \quad \square$$

Since every subgraph of a bipartite graph is bipartite, one can derive an upper bound for the density of the partition. For each class X of P^*, we have $d(X) \leq \frac{|X|}{4}$ and since P^* is a partition of the vertex set we have the following property.

Property 5. *Let $G = (V_1 \cup V_2, E)$ be a bipartite graph and P^* an optimal partition of G, then the following inequality holds:*

$$d(P^*) = \sum_{X \in P^*} d(X) \leq \sum_{X \in P^*} \frac{|X|}{4} = \frac{n}{4}.$$

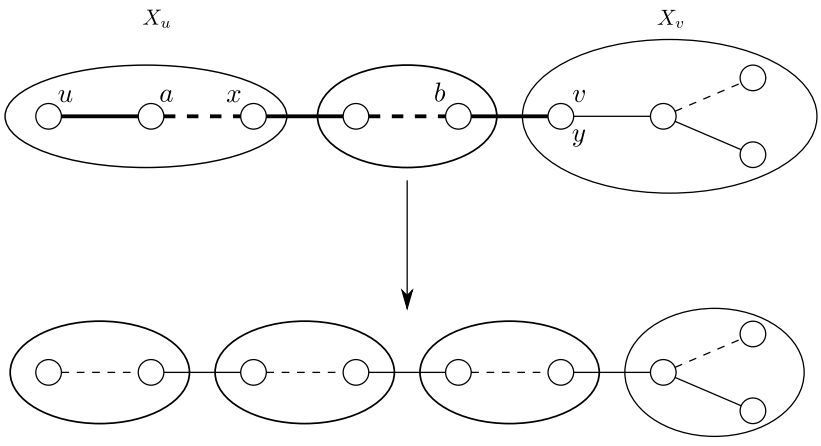


Fig. 3. Example of an augmenting chain (in bold) and a partition transformation on a tree with 5 edges in C and $|X_u| = 3$, $|X_v| = 4$, $v = y$ and $u \neq x$. Each class of P is represented by circled vertices; the edges of M are represented by dashed lines.

Notice that the bound is tight if G admits a perfect matching. It shows a link between the classes of P^* and a perfect matching of a bipartite graph. In the case of trees the link is stronger as stated in the next lemma.

Lemma 5. *Let T be a tree, M^* a maximum matching of T and P^* an optimal partition of T. Then $|M^*| = |P^*|$.*

Proof. The inequality $|M^*| \geq |P^*|$ comes from Corollary 1. Indeed one can construct a matching M by choosing an edge in each class of P^*. Now we show that M is a maximum matching. Suppose by contradiction that M is not a maximum matching. Then by the Petersen Theorem [24] there exists an augmenting path. Let C be a minimum (in the number of edges) augmenting path and u, v its extremities.

We show that u and v cannot be in the same class. If $C \neq (u, v)$, then C has a length greater than 3 and u and v belong to different classes according to Lemma 4. If $C = (u, v)$ since u and v are not covered by the matching, they belong to different classes otherwise it contradicts Corollary 1. Furthermore, their classes contain at least 3 vertices, otherwise u and v would be covered by the matching. Let X_u (resp. X_v) be the class of u (resp. v) and let $x \in X_u$ (resp. $y \in X_v$) be the closest vertex to v (resp. u) on C (note that x could be u and y could be v). Since x and y belong to different classes then $x \neq y$. Let C_{xy} be the subpath of C from x to y. We denote by a (resp. b) the neighbor of u (resp. v) on C (see Fig. 3).

Since C is minimum, every vertex of $C \setminus \{a, b\}$ has a degree at most two in the subgraph induced by its class. Indeed u (resp. v) is not saturated by the matching and since X_u (resp. X_v) is a star, the degree of u (resp. v) in X_u (resp. X_v) is one. If the degree of at least one of the remaining vertices is greater than two then there exists a smaller augmenting path which is in contradiction with the minimality of C.

Let M' be the matching obtained from M by exchanging the edges on the augmenting path. Now we create a new partition P' from P^* by removing x from X_u and y from X_v. Each edge of C not in M forms a new class of P'. Notice that the neighbors of a (resp. b) which were in X_a (resp. X_b) and that do not belong to C are in the new class of a (resp. b).

Since (u, a) and (v, b) are in M' and since $u \neq v$, we have $a \neq b$ and X'_a the class of a in P' is different from X'_b the class of b in P'. Thus all classes of the partition P' are stars.

During this process, X_u and X_v lose one vertex and a new class is created. Thus $\Delta = d(P') - d(P^*) = \frac{1}{2} - \frac{1}{|X_u|\ (|X_u|-1)} - \frac{1}{|X_v|\ (|X_v|-1)}$ with $|X_u|$ and $|X_v|$ greater than or equal to 3. Thus $\Delta > 0$ and P^* is not optimal. □

Using some classical results in graph theory we have the following corollary.

Corollary 2. *Let T^* be a minimum vertex cover of a tree and P^* an optimal partition of the same tree. Then each class of P^* contains exactly one vertex of T^*.*

Proof. From Corollary 1 we know that each class of P^* is a non-trivial star. Thus each class of P^* contains at least a vertex of T^*. From Lemma 5 and König's theorem [19] we get

$$|P^*| = |M^*| = |T^*|.$$

Then each class of P^* contains exactly one vertex of T^*. □

Let us now derive a polynomial algorithm for the special case of trees. Let T be a rooted tree in u and let T_i be the subtree induced by i a child of u and its descendants. Let F'_i be the forest induced by the vertices of $T_i \setminus \{i\}$, see Fig. 4. The basic idea is to construct the optimal partition of T by a recursive construction using the optimal partition of T_i and F'_i. This algorithm gives the following theorem.

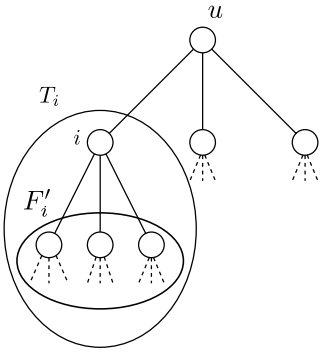


Fig. 4. A rooted tree in u with a child i, a subtree T_i and the forest F'_i.

Theorem 3. *The problem* Max Dense Graph Partition *is polynomial on trees.*

Proof. Let T^* be a minimum vertex cover on T and $W \subseteq V$ be the set of the children of u. Remark that the computation of T^* is polynomial on trees [24].

Suppose that $u \in T^*$, then u has two types of children: $W' = W \setminus T^*$ and $W \cap T^*$. By Corollary 2, u and another vertex of T^* cannot be in the same class in an optimal partition of T. For the children of u that are in T^* we use the optimal partition of their subtrees.

Thus we only consider the children of u that are in W' and that are not isolated (isolated vertices have to be in the class of u). Let Δ_i be the difference between the value of the optimal partition of T_i and the value of the optimal partition of F'_i. It is clear that $\Delta_i > 0$. We create an order on W' defined by $i \prec j$ if $\Delta_i < \Delta_j$. Let X_u be the class containing u in P^* an optimal partition of T. If $j \in X_u$ then $\forall i \in W'$ such that $i \prec j$ we have $i \in X_u$, otherwise by exchanging i and j in X_u one can create a new partition P' of T and $d(P') - d(P^*) = \Delta_j - \Delta_i > 0$ and P^* is not optimal. The class X_u can be constructed by adding each $i \in W'$ using the order $\prec$ until $d(X_u \cup \{j\}) - d(X_u) < \Delta_j$. The optimal partition of T is obtained by the class X_u, the optimal partition of F'_i for each $i \in X_u \setminus \{u\}$ and the optimal partition of T_j for each j a child of u that is not in X_u.

Now suppose that $u \notin T^*$ then all the children of u are in T^*. By Corollary 2, u must be in the class of one of its children. For each i a child of u, consider the tree T_i^u consisting of T_i, the vertex u and an edge between u and i. Using the same argument as in the previous paragraph one can obtain its optimal partition. The optimal partition of T is obtained by adding u to the class of its child that leads to the optimal partition and by taking the optimal partition of the tree T_i for all the other children i of u.

Finally, one can obtain the optimal partition of F'_u with the optimal partition of each T_i. By applying this procedure from the leaves to the root, one can obtain the optimal partition of a rooted tree T. □

5. Conclusion

In this paper we presented some hardness results on maximizing the density of a vertex partition. We showed that this problem is equivalent to minimizing the sparsity of a vertex partition. Theorem 1 states that these two problems are NP-hard and Theorem 2 gives a non-approximability result for the minimization of the sparsity. Due to the strong link with the proper graph coloring problem in the reduction, it should be interesting to study Min Sparse Graph Partition on special classes of graph for which the graph coloring problem is easy. In Section 4, we give a polynomial time algorithm for Max Dense Graph Partition on trees. The next step would be the extension of these results to bipartite graphs. From a more practical point of view some empirical tests could be done to study the behavior of the density in the context of community detection.

References

[1] D. Aloise, A. Deshpande, P. Hansen, P. Popat, NP-hardness of euclidean sum-of-squares clustering, Machine Learning 75 (2) (2009) 245–248.
[2] G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, V. Kann, Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.
[3] E. Boros, P.L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, I. Muchnik, An implementation of logical analysis of data, IEEE Transactions on Knowledge and Data Engineering 12 (2) (2000) 292–306.
[4] U. Brandes, A faster algorithm for betweenness centrality, Journal of Mathematical Sociology 25 (2001) 163–177.
[5] M. Brusco, S. Stahl, Branch-and-Bound Applications in Combinatorial Data Analysis, Springer, 2005.
[6] M. Charikar, Greedy approximation algorithms for finding dense components in a graph, in: APPROX, 2000, pp. 84–95.
[7] A. Clauset, C. Moore, M.E.J. Newman, Hierarchical structure and the prediction of missing links in networks, Nature 453 (2008) 98–101.
[8] J. Darlay, Analyse Combinatoire de données: Structure et Optimisation. Ph.D. Thesis, Université de Grenoble, 2011.
[9] S. Dasgupta, The hardness of k-means clustering. Technical Report, University of California, 2008.
[10] I. Dhillon, Y. Guan, B. Kulis, Weighted graph cuts without eigenvectors: a multilevel approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (11) (2007) 1944–1957.
[11] R. Diestel, Graph Theory, in: Graduate Texts in Mathematics, vol. 173, Springer-Verlag, Heidelberg, 2005.

[12] T. Feder, P. Hell, S. Klein, R. Motwani, List partitions, SIAM J. Discrete Math. 16 (3) (2003) 449–478.
[13] U. Feige, G. Kortsarz, D. Peleg, The dense *k*-subgraph problem, Algorithmica 29 (1999) 2001.
[14] S. Fortunato, Community detection in graphs, Physics Reports 486 (3–5) (2010) 75–174.
[15] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences), W.H. Freeman & Co. Ltd., 1979.
[16] A.V. Goldberg, Finding a maximum density subgraph. Technical Report UCB/CSD-84-171, EECS Department, University of California, Berkeley, 1984.
[17] N. Gulbahce, S. Lehmann, The art of community detection, BioEssays 30 (2008) 934–938.
[18] J. Kleinberg, An impossibility theorem for clustering, Advances in Neural Information Processing Systems (2002).
[19] D. König, Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre, Mathematische Annalen 77 (1916) 453–465 (in German).
[20] M. Maravalle, B. Simeone, R. Naldini, Clustering on trees, Computational Statistics & Data Analysis 24 (2) (1997) 217–234.
[21] M.E.J. Newman, Detecting community structure in networks, The European Physical Journal B — Condensed Matter and Complex Systems 38 (2) (2004) 321–330.
[22] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E 69 (2004) 026113.
[23] S. Schaeffer, Graph clustering, Computer Science Review 1 (1) (2007) 27–64.
[24] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Springer, 2003.
[25] J.P. Scott, Social Network Analysis: A Handbook, SAGE Publications, 2000.
[26] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[27] J. Síma, S.E. Schaeffer, On the NP-completeness of some graph cluster measures, in: SOFSEM 2006: Theory and Practice of Computer Science, in: Lecture Notes in Computer Science, vol. 3831, 2006, pp. 530–537.
[28] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 1st edition, in: The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, 1999.