Noname manuscript No.
(will be inserted by the editor)

Structural and algorithmic properties of 2-community structures

**Cristina Bazgan · Janka Chlebikova ·
Thomas Pontoizeau**

Abstract We investigate the structural and algorithmic properties of 2-community structures in graphs introduced recently by Olsen [18]. A 2-community structure is a partition of a vertex set into two parts such that for each vertex the numbers of neighbours in/outside its own part and the sizes of the parts are correlated. We show that some well studied graph classes as graphs of maximum degree 3, minimum degree at least $|V| - 3$, trees and also others, have always a 2-community structure. Furthermore, a 2-community structure can be found in polynomial time in all these classes, even with additional request of connectivity in both parts. We introduce a concept of a weak 2-community and prove that in general graphs it is NP-complete to find a balanced weak 2-community structure with or without request for connectivity in both parts. On the other hand, we present a polynomial-time algorithm to solve the problem (without the condition for connectivity of parts) in graphs of degree at most 3.

Keywords: graph theory, complexity, graph partitioning, community structure, clustering, social networks

Cristina Bazgan
Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016 Paris, France
E-mail: bazgan@lamsade.dauphine.fr

Janka Chlebikova
School of Computing, University of Portsmouth, UK
E-mail: janka.chlebikova@port.ac.uk

Thomas Pontoizeau
Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016 Paris, France
E-mail: thomas.pontoizeau@lamsade.dauphine.fr

1 Introduction

The research around community structures can be seen as a contribution to the well-establish research of clustering and graph partitioning. The partition of graphs have been intensively studied with various measures to evaluate their quality, see e.g. [2,7,14,17,19] for an overview.

A standard abstract model for any kind of social networks such as Facebook or Linkedin is a graph, in which vertices are members of the network and edges are relationships between members. In such model 'a community' intuitively corresponds to a subgraph that has 'more relationships' inside the subgraph than outside of it. More generally, 'a community structure' corresponds to a partition of a graph into communities.

There have been several attempts to define the concept of communities formally, a good introduction including the motivation can be found in [1,6, 11,18,20]. One of the first definitions of a community was motivated by the searching links in web graphs and introduced by Flake et al. [13]. It defines a community as a set of vertices C such that each vertex in C has at least as many neighbours inside C as outside. The same notion called an 'alliance in graphs' were introduced by Kristiansen et al. [16] and investigated further in various papers. The concept of communities and community structures have received a significant attention in further research where also some modified definitions of communities were studied e.g. the difference between the number of outside and inside neighbours should be larger than a given constant, the community should also be a dominating set, see e.g. [4,5,15] for overview and further references.

In this paper we study the structural and complexity problems of the recent definition of a community structure that reflects the sizes of communities too [10,11,18]. This new approach to communities is supported by the practical experiments showing the importance of capturing the sizes of communities for a better description of their properties [18].

The general concept of a community structure does not put any restriction on the number of communities. This paper focuses on a partition with two communities where the problems are already appealing. The presented techniques offer some possibilities for an extension to a larger number of communities. Informally, a 2-community structure is a partition of the vertex set into two parts A, B such that for each vertex, say from part A, the ratio 'the number of neighbours in part A' over the size of A (excluding the vertex itself) is at least as large as 'the number of neighbours in part B' over the size of B. To generalise, in a k-community structure, the ratio must be valid for every two communities. We also introduce a weak community structure in which the vertex itself contributes to the ratio. The ratio condition in the latter definition is weaker, but it reflects the reasonable requirement that each member should be considered as a part of its own community (see Section 2 for the technical details). Even if there are minor differences between the definitions, the structural and complexity results for the two problems are very different as it is presented in this paper. Both definitions are relevant to describe the community structures, the choice depends on the suitability of the model.

We also study the 2-communities problems with additional constraints such as connectivity or equality of sizes for both parts (a balanced partition). The connectivity request corresponds to the essential condition that each member in the community should 'indirectly know' all members in its own community, where the 'indirectly know' relation corresponds to a path between two vertices in the graph. The study of balanced communities is motived by the practical interest for equal size of the communities. In general, the balanced graph partitions are well studied, e.g. due to its applications in the divide-and-conquer algorithms, see e.g. [8]. In the balanced partition problem, which can be seen as a generalisation of the bisection problem to any given number of parts, the goal is to minimise the number of edges between partitions. It is known that the problem cannot be approximated within any finite factor in polynomial time in general graphs and it remains APX-hard even on trees of constant maximum degree [12]. It demonstrates that some graph partitions problems that are related to e.g. balanced communities are hard to solve even for restricted graph classes and indicates hardness of various problems related to a community structure too. Hence all positive results in community structure problems would be important to get better understanding of the differences between community and partition problems.

Furthermore, a community structure is in fact a graph partition with a restricted number of edges between parts, therefore the new results for communities may find applications in the areas similar to a graph partition such as parallel-computing, VLSI-circuit design, route planning [9] and divide-and-conquer algorithms [21].

There are only a few results related to this new definition of a community. Olsen [18] proved that a community structure (without the condition on the exact number of communities) can be found in polynomial time in any graph with at least 4 vertices, except a star. Recently, Estivill-Castro *et al.* [11] claimed that the problem to find a k-community structure with restriction to all communities to be connected and equal size is NP-complete in general graphs, but polynomially solvable in trees. In [18] Olsen also proved that it is NP-complete to decide, whether there is a community structure in a graph in which a given set of vertices is included in a community.

Our contribution

The following overview summarises our results achieved in this paper. All considered graphs are of size at least 4 and are not stars. If a 2-community structure with certain properties exists for a graph class, then it exists for all the graphs from the class.

(i) *trees:*
   – a connected 2-community structure exists and can be found in linear time (Theorem 1),
   – there are trees with a balanced 2-community structure, but without a connected balanced weak 2-community structure (Remark 2),

(ii) *graphs of maximum degree* 3*:*
– a connected 2-community structure exists and can be found in polynomial time (Theorem 2),
– a balanced weak 2-community structure exists and can be found in polynomial time (Theorem 6),
– there are graphs without a balanced 2-community structure (Remark 1),
– there are graphs with a balanced 2-community structure, but without a connected balanced weak 2-community structure (Remark 2)

(iii) *graphs of minimum degree* $(|V| - 3)$*, complements of bipartite graphs, graphs with minimum degree* $\lceil \frac{(c-1)\cdot|V|}{c} \rceil$ *where c is the size of an inclusion-wise maximal clique in the graph:*
– a connected 2-community structure exists and can be found in polynomial time (Theorems 3, 4, 5)

(iv) *graphs of bounded tree-width:*
– there are graphs without a balanced 2-community structure (Remark 1), but to decide whether such a structure exists and if it exists, find it, can be done in polynomial time (Remark 3)

Estivill-Castro *et al.* [10] proved that the problem of finding a balanced 2-community structure is NP-complete in general graphs. In Section 4 we show that the same result also holds for a weak community, even with additional constraint of connectivity for both parts. We also present a shorter proof of the known NP-complete result for a balanced 2-community structure in general graphs based on an alternative definition of community structure [4], which also implies NP-completeness for a connected balanced 2-community structure.

The paper is structured as follows. In Section 2 we introduce formally some notations and definitions of studied problems. In Section 3 we show that in some well-studied graph classes a 2-community structure always exists and can be found in polynomial time, even with additional request for connectivity in both parts. In Section 4 we focus on the balanced 2-community structure and present the structural and algorithmic results in general graphs and some graph classes. Conclusions and open problems are provided in Section 5.

2 Preliminaries

In the paper, all considered graphs are simple, undirected and connected. Let $G = (V, E)$ be a graph. For a vertex $v \in V$, let $d(v)$ be the degree of the vertex v and for any subgraph H of the graph G let $N_H(v)$ be the set of the neighbours of v in H, $N_H[v] = N_H(v) \cup \{v\}$ and let $d_H(v) = |N_H(v)|$. For a given partition of V into two parts (a 2-*partition*), let an *in-neighbour* of v (resp. *out-neighbour*) be a neighbour in its own part (resp. out of its part) and $d_{in}(v)$ (resp. $d_{out}(v)$) denote the number of in-neighbours of v (resp. out-neighbours). For a graph G and a subset of vertices $S \subseteq V$, let $G[S]$ denote the subgraph of G induced by S. A partition $\{C_1, C_2\}$ of V is connected if the subgraphs $G[C_1]$ and $G[C_2]$ are connected and it is balanced if the sizes of C_1

and C_2 differ by at most 1. The *cut size* of a 2-partition is the number of edges that have end vertices in the different parts of the partition. A graph is said to be of minimum (resp. maximum) degree k if any vertex of the graph has degree at least (resp. at most) k. A *pendant vertex* of G is any vertex of degree 1. A *star* is a complete bipartite graph $K_{1,\ell}$ for any $\ell \geq 1$. The complement graph $\overline{G} = (V, \overline{E})$ of a graph $G = (V, E)$ is the graph in which $\{u, v\} \in E$ iff $\{u, v\} \notin \overline{E}$ for all vertices $u, v \in V$. A graph G is 2-*colourable* if there exists a partition $\{C_1, C_2\}$ of V such that $G[C_1]$, $G[C_2]$ contain only isolated vertices.

Now we introduce Olsen's definition of a k-community structure from [18].

Definition 1 A *k-community structure* for a connected graph $G = (V, E)$ is a partition $\varPi = \{C_1, \ldots, C_k\}$ of V, $k \geq 2$, such that $\forall i \in \{1, \ldots, k\}, |C_i| \geq 2$, and $\forall v \in C_i, \forall C_j \in \varPi$, $j \neq i$, the following holds

$$\frac{|N_{C_i}(v)|}{|C_i| - 1} \geq \frac{|N_{C_j}(v)|}{|C_j|}$$

For a *weak k-community structure*, the condition above is replaced by a "weaker" condition

$$\frac{|N_{C_i}[v]|}{|C_i|} \geq \frac{|N_{C_j}(v)|}{|C_j|}.$$

Notice that a k-community structure is obviously a weak k-community structure since $\frac{|N_{C_i}[v]|}{|C_i|} = \frac{|N_{C_i}(v)|+1}{|C_i|-1+1} \geq \frac{|N_{C_i}(v)|}{|C_i|-1}$, but the opposite is not true (see Fig. 1).

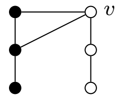

Fig. 1 A weak 2-community structure of a graph (presented by the colours black and white) in which the vertex v does not satisfy the condition of a 2-community structure but satisfies the condition of a weak 2-community structure from Definition 1.

In this paper we investigate a community structure for a fixed number of two communities and also study some variants of the 2-Community problem:

2-Community
Input: A graph $G = (V, E)$.
Question: Does G have a 2-community structure?
It means, is there a 2-partition $\{C_1, C_2\}$ of the vertex set V such that $|C_1|, |C_2| \geq 2$, and for each vertex $v \in C_i$, $i \in \{1, 2\}$,

$$\frac{|N_{C_i}(v)|}{|C_i| - 1} \geq \frac{|N_{C_{3-i}}(v)|}{|C_{3-i}|} \tag{1}$$

Obviously, if G has a 2-community structure, it must have at least 4 vertices and be non-isomorphic to a star which we assume in the paper even without explicitly mentioning that in some informal parts.

In the Weak 2-Community problem we are looking for a weak 2-community structure in a graph where the condition (1) is replaced by

$$\frac{|N_{C_i}[v]|}{|C_i|} \geq \frac{|N_{C_{3-i}}(v)|}{|C_{3-i}|}. \tag{2}$$

Adding the balanced condition to the 2-Community problem, we obtain the Balanced 2-Community problem introduced by Estivill-Castro *et al.* [10]. Similarly we can define the Balanced Weak 2-community problem.

The additional constraint which asks for subgraphs induced by each part of the partition to be connected is a natural condition useful for the problems related to the connectedness. The Connected 2-Community problem is to decide if a graph has a connected 2-community structure, i.e. a 2-community structure $\{C_1, C_2\}$ such that the subgraphs induced by C_1, C_2 are connected. We can define analogous problems for weak and balanced versions.

3 Connected 2-community structures in some graph classes

In this section we show that if a graph has certain structural properties, then it has a connected 2-community structure which can be found in polynomial time. More precisely, we prove that such a statement is valid for trees and graphs of high minimum or low maximum degrees.

Theorem 1 *Every tree with at least* 4 *vertices (except a star) has a connected* 2*-community structure that can be found in linear time.*

Proof Let $G = (V, E)$ be a tree not isomorphic to a star. We prove that there exists an edge $e \in E$ such that two connected components of $G \setminus e$ form a 2-partition which is a connected 2-community structure.

Let $e = \{u, v\}$ be an edge in E such that $d(v)$, $d(u) \geq 2$ (due to the assumption about G such an edge e must exist). Consider a partition $\{X_u, X_v\}$ of V with X_u (resp. X_v) be the set of vertices of the connected component of $G \setminus e$ containing u (resp. v).

First we notice that only one of the vertices u and v may not satisfy the condition (1). If this is not true then $\frac{d(u)-1}{|X_u|-1} < \frac{1}{|X_v|}$ and $\frac{d(v)-1}{|X_v|-1} < \frac{1}{|X_u|}$. Since $d(u), d(v) \geq 2$, it implies $|X_v| < \frac{|X_u|-1}{d(u)-1} \leq |X_u| - 1$ and $|X_u| < \frac{|X_v|-1}{d(v)-1} \leq |X_v| - 1$, which is not possible.

If both vertices u and v satisfy the condition (1), then $\{X_u, X_v\}$ is obviously a 2-community structure. If not, then without loss of generality, let the vertex u satisfy the condition (1) and v do not. Then the Update procedure is repeated and if no update is possible, a modified partition $\{X_u, X_v\}$ is already a 2-community structure as it is shown later.

The Update procedure:

Let $v_1, v_2, \ldots, v_{d(v)-1}$ be the neighbours of v excluding u (there is at least one such a vertex due to our assumption $d(v) \geq 2$). For each i, $1 \leq i \leq d(v)-1$, and $e_i = \{v, v_i\} \in E$, let X_i be the set of vertices of the connected component in $G \setminus e_i$ containing v_i.

Notice that if for all j, $1 \le j \le d(v) - 1$, $d(v_j) = 1$, then v must already satisfy the condition (1) in the partition $\{X_u, X_v\}$ at the beginning of the Update procedure.

Hence from now we suppose that v has at least one neighbour of degree at least 2 excluding u. In the following we show that there exists j, $1 \le j \le d(v) - 1$, such that $d(v_j) > 1$ and the vertex v satisfies the condition (1) in the partition $\{X_j, V \setminus X_j\}$. Indeed, suppose that for all j, $1 \le j \le d(v) - 1$, with $d(v_j) > 1$, this is not true. Notice that for each such j and the partition $\{X_j, V \setminus X_j\}$ must hold $\frac{d(v)-1}{n-|X_j|-1} < \frac{1}{|X_j|}$ which implies that

$$d(v)|X_j| < n - 1. \tag{3}$$

Moreover, for any j, $1 \le j \le d(v) - 1$ with $d(v_j) = 1$ we have $|X_j| = 1$ and hence

$$d(v)|X_j| < n - 1, \tag{4}$$

since G is not a star. Recall that v doesn't satisfy the condition (1) in the partition $\{X_u, X_v\}$, hence $\frac{d(v)-1}{|X_v|-1} < \frac{1}{|X_u|}$ and also

$$d(v)|X_u| < n - 1, \tag{5}$$

Summing (3), (4) and (5) together, we obtain $d(v) \sum_{j=1}^{d(v)} |X_j| = d(v)(n - 1) < d(v)(n - 1)$, a contradiction.

Hence, there exists i, $1 \le i \le d(v) - 1$ such that $d(v_i) > 1$ and the vertex v satisfies the condition (1) in the partition $\{X_i, V \setminus X_i\}$. Then, relabel $u := v$ and $v := v_i$ and return to the beginning of the Update procedure.

Each time the labels of u and v are updated, the size of X_u strictly increases by at least one, hence the whole process always terminates. A final partition at the end of the process is a connected 2-community structure because both partitions correspond to two connected components of a tree obtained by removing an edge.

Notice that finding such an edge can be done in $O(|V|)$ operations. First, in constant time fix an edge $e = \{u, v\}$ such that $d(v), d(u) \ge 2$. Then, consider $G \setminus e$ as a union of two trees T_u and T_v, where T_u is a tree on the vertex set X_u rooted in u (and similarly for T_v on X_v rooted in v). For each vertex w of G calculate recursively the size of the subtree of T_u (or T_v) rooted in w which can be done in time $O(|V|)$. Finally, using the sizes of the subtrees, check if $\{X_u, X_v\}$ corresponds to a 2-community structure and if needed, update X_u, X_v according to the algorithm. The number of such updates is clearly at most $|E|$. Since G is a tree, the repetition of the Update procedure finishes with a connected 2-community structure in $O(|V|)$ time. □

Very recently, Estivill-Castro *et al.* proved in [11] the same result using different methods. Our approach is more structural and the proof for the existence of an edge that connects two communities results directly in a linear time algorithm.

Now we investigate graphs that may contain cycles but that still have low densities, namely the graphs of maximum degree 3. First, the restrictions on the size of partitions are discussed to ensure the vertices fulfil the condition (1) of a 2-community structure.

Lemma 1 *Let $G = (V, E)$ be a graph of maximum degree 3 of size n. Let $\{C_1, C_2\}$ be a partition of V such that $\lceil \frac{n-1}{3} \rceil \leq |C_i| \leq n - \lceil \frac{n-1}{3} \rceil$, $i = 1, 2$. Then each vertex of degree 3 in G with at most one out-neighbour fulfils the condition (1) of a 2-community structure.*

Furthermore, if for some $i \in \{1, 2\}$, $|C_i| = \lceil \frac{n-1}{3} \rceil$ (or also $|C_i| = \lceil \frac{n-1}{3} \rceil + 1$ in case $n \equiv 1 \mod 3$) then each vertex of degree 3 in C_i with two out-neighbours fulfils the condition (1) too.

Proof Let $\{C_1, C_2\}$ be a fixed partition of G such that $\lceil \frac{n-1}{3} \rceil \leq |C_i| \leq n - \lceil \frac{n-1}{3} \rceil$, $i = 1, 2$. It is clear that the condition (1) is true for each vertex which has only neighbours in its own part. Firstly, suppose the vertex v from C_i, $i \in \{1, 2\}$ has exactly one out-neighbour.

Since $|C_i| \leq n - \lceil \frac{n-1}{3} \rceil$, then obviously $|C_i| \leq n - \frac{n-1}{3}$ and $\frac{2}{|C_i|-1} \geq \frac{1}{n-|C_i|}$. Therefore the condition (1) is fulfilled for the vertex v.

Now suppose that for $i \in \{1, 2\}$ there is a vertex $v \in C_i$ with exactly two out-neighbours and $|C_i| = \lceil \frac{n-1}{3} \rceil$. Obviously, $\lceil \frac{n-1}{3} \rceil \leq \frac{n+2}{3}$ and hence $2\lceil \frac{n-1}{3} \rceil - 2 \leq n - \lceil \frac{n-1}{3} \rceil$ which implies $\frac{1}{\lceil \frac{n-1}{3} \rceil - 1} \geq \frac{2}{n - \lceil \frac{n-1}{3} \rceil}$. This corresponds to the condition (1) for the vertex v. Similarly if $|C_i| = \lceil \frac{n-1}{3} \rceil + 1$ and $n \equiv 1 \mod 3$: $n - 1 = 3\lceil \frac{n-1}{3} \rceil$ which implies $\frac{1}{\lceil \frac{n-1}{3} \rceil} \geq \frac{2}{n - \lceil \frac{n-1}{3} \rceil - 1}$. □

Lemma 2 *Let $G = (V, E)$ be a graph of maximum degree 3 of size n. Let $\{C_1, C_2\}$ be a partition of V such that $\lceil \frac{n-1}{3} \rceil \leq |C_1| \leq \lfloor \frac{n}{2} \rfloor$. Then each vertex of degree 2 in C_1 with at most one out-neighbour fulfils the condition (1) of a 2-community structure.*

If the partition is balanced, then each vertex of degree 2 in G with at most one out-neighbour fulfils the condition (1).

Proof Let $\{C_1, C_2\}$ be a partition of V such that $\lceil \frac{n-1}{3} \rceil \leq |C_1| \leq \lfloor \frac{n}{2} \rfloor$. Obviously, any vertex of degree 2 with no neighbours out of its own part fulfils the condition (1). Moreover any vertex of degree 2 in C_1 with only one out-neighbour satisfies $\frac{1}{|C_1|-1} \geq \frac{1}{|C_2|}$ since $|C_1| \leq |C_2|$.

If the partition is balanced, then $\frac{1}{|C_1|-1} \geq \frac{1}{|C_2|}$ and $\frac{1}{|C_2|-1} \geq \frac{1}{|C_1|}$, and hence the vertices of degree 2 from both parts with exactly one out-neighbour satisfy the condition (1). □

Lemma 3 *Let $G = (V, E)$ be a graph of maximum degree 3 of size n and $\{C_1, C_2\}$ be a partition of V such that $\lceil \frac{n-1}{3} \rceil \leq |C_i| \leq n - \lceil \frac{n-1}{3} \rceil$, $i = 1, 2$.*

If the partition has one of the properties (i)-(iii) where only specified vertices may have out-neighbours (the other ones have only in-neighbours), then $\{C_1, C_2\}$ is a 2-community structure on G:

(i) The vertices of degree 2 from the smaller part and all the vertices of degree 3 have at most one out-neighbour.

(ii) *The vertices of degree* 2 *and* 3 *have at most one out-neighbour and the partition is balanced.*

(iii) *The vertices of degree* 2 *from the smaller part have at most one out-neighbour, the vertices of degree* 3 *in* C_i*, for some* $i \in \{1, 2\}$*, have at most two out-neighbours and* $|C_i| = \lceil \frac{n-1}{3} \rceil$ *(or also* $|C_i| = \lceil \frac{n-1}{3} \rceil + 1$ *if* $n \equiv 1 \mod 3$*) and the vertices of degree* 3 *in* C_{3-i} *have at most one out-neighbour.*

Proof In each case (i), (ii), or (iii), all the vertices of the graph G satisfy the condition (1) due to Lemmas 1 and 2. Hence, $\{C_1, C_2\}$ is a 2-community structure on G. □

Lemma 4 *Every connected graph of maximum degree* 3 *on* n *vertices,* $n \geq 4$*, (except a star) has a connected partition* $\{C_1, C_2\}$ *such that* $\lceil \frac{n-1}{3} \rceil \leq |C_i| \leq n - \lceil \frac{n-1}{3} \rceil$*,* $i = 1, 2$*. Moreover, such a partition can be found in polynomial time.*

Proof Let $G = (V, E)$ be a graph with the given properties. If G is a tree, take a pendant vertex $u \in V$ and let $v \in V$ be its neighbour. If G is not a tree, let $\{u, v\}$ be an edge of a cycle in G. Since G is not isomorphic to a star such an edge must exist.

Initially, put into C_1 the vertices u, v together with their pendant vertices, if it is applicable. If there is a vertex z of degree 2 adjacent to u and v, update $C_1 := C_1 \cup \{z\}$. Define $C_2 := V \setminus C_1$.

The algorithm keeps connectivity of $G[C_1]$ and $G[C_2]$ and extends C_1 either by transferring vertices from C_2 to C_1 or relabelling a suitable connected part of the graph until $\lceil \frac{n-1}{3} \rceil \leq |C_i| \leq n - \lceil \frac{n-1}{3} \rceil$, $i = 1, 2$.

The algorithm starts with the initial set C_1 and repeats the Update Procedure until $|C_1| \geq \lceil \frac{n-1}{3} \rceil$. In each run of the procedure only one of the options 1 or 2 is executed.

The Update procedure:
Let w be a vertex in C_2 which has a neighbour in C_1 (such a vertex must exist since G is connected).
Option 1: If the subgraph induced by $C_2 \setminus \{w\}$ is connected, put

$$C_1 := C_1 \cup \{w\}, C_2 := C_2 \setminus \{w\}.$$

Option 2: If the subgraph induced by $C_2 \setminus \{w\}$ is disconnected (w must be of degree 3), then denote by A, B the vertex-sets of two connected induced subgraphs of G on $C_2 \setminus \{w\}$. Depending on the size of A, the following update is executed.

- If $|A| \leq n - 2\lceil \frac{n-1}{3} \rceil$, put

$$C_1 := C_1 \cup A \cup \{w\}, C_2 := B.$$

  Notice that $|C_1| \leq n - \lceil \frac{n-1}{3} \rceil$, $\{C_1, C_2\}$ is a connected partition and the size of C_1 strictly increased.

- If $n - 2\lceil\frac{n-1}{3}\rceil + 1 \leq |A| \leq n - \lceil\frac{n-1}{3}\rceil$, then notice that $|A| \geq \lceil\frac{n-1}{3}\rceil$ and put
$$C_1 := A,\ C_2 := V \setminus A.$$
Obviously, $\{C_1, C_2\}$ is a connected partition with $\lceil\frac{n-1}{3}\rceil \leq |C_i| \leq n - \lceil\frac{n-1}{3}\rceil$, $i = 1, 2$, hence the Update Procedure halts.
- If $|A| > n - \lceil\frac{n-1}{3}\rceil$, put
$$C_1 := C_1 \cup B \cup \{w\},\ C_2 := A.$$
Notice that $|C_1| < \lceil\frac{n-1}{3}\rceil$, $\{C_1, C_2\}$ is a connected partition and the size of C_1 strictly increased.

If $|C_1| \geq \lceil\frac{n-1}{3}\rceil$ after the execution of the option 1 or 2, then the Update procedure halts, otherwise the Update Procedure is repeated again.

By our construction, the partition $\{C_1, C_2\}$ remains connected during each run of the Update procedure

Each time the Update procedure is executed, the size of C_1 strictly increases, hence the algorithm always terminates.

At the end of the algorithm $\lceil\frac{n-1}{3}\rceil \leq |C_i| \leq n - \lceil\frac{n-1}{3}\rceil$, $i = 1, 2$ and the algorithm clearly runs in a polynomial time □

Theorem 2 *Every connected graph of maximum degree* 3 *with at least* 4 *vertices (except a star) has a connected* 2*-community structure which can be found in polynomial time.*

Proof Let $G = (V, E)$ be a connected graph of maximum degree 3 on n vertices, $n \geq 4$, not isomorphic to a star. Due to Lemma 4, a connected partition $\{C_1, C_2\}$ of V such that $\lceil\frac{n-1}{3}\rceil \leq |C_i| \leq n - \lceil\frac{n-1}{3}\rceil$, $i = 1, 2$, can be found in polynomial time. Let $\{C_1, C_2\}$ be such a partition and notice that the vertices that do not satisfy the condition (1) can be split into two categories:

(A) if there exists $i \in \{1, 2\}$ such that $|C_i| > \lceil\frac{n-1}{3}\rceil$ in case $n \not\equiv 1 \mod 3$ or $|C_i| > \lceil\frac{n-1}{3}\rceil + 1$ in case $n \equiv 1 \mod 3$, then all the vertices of degree 3 in C_i with two out-neighbours,

(B) if the partition is not balanced, then all the vertices of degree 2 in the larger part with one out-neighbour.

The algorithm starts with the initial partition $\{C_1, C_2\}$ and then the Improvement Procedure (consisting in three stages) can be applied several times. The procedure transfers step-by-step all the vertices of degree at least 2 (with exactly one neighbour in its own part) between C_1 and C_2 or relabel the sets, until all the vertices satisfy the condition (1). Since the initial partition is connected, transferring vertices with such a property never disconnects any part of the partition.

The Improvement Procedure: Stage 1 (Category (A) vertices)
In this stage we handle vertices in C_2 of degree 3 with two out-neighbours by

transferring them into C_1, keeping the size of C_1 smaller than $n - \lceil \frac{n-1}{3} \rceil$ and ensuring connectivity of the partition $\{C_1, C_2\}$.

While $|C_1| < n - \lceil \frac{n-1}{3} \rceil$ *and there is a vertex* $u \in C_2$ *with two out-neighbours, update*

$$C_1 := C_1 \cup \{u\}, \quad C_2 := C_2 \setminus \{u\}.$$

Notice that each iteration of Stage 1 decreases the size of the cut by at least one.

The Improvement Procedure: Stage 2 (Category (A) vertices)
Similarly to Stage 1, in Stage 2 we handle vertices in C_1 of degree 3 with two out-neighbours by transferring them into C_2, keeping the size of C_2 smaller than $n - \lceil \frac{n-1}{3} \rceil$ and ensuring connectivity of the partition $\{C_1, C_2\}$.

While $|C_2| < n - \lceil \frac{n-1}{3} \rceil$ *and there is a vertex* $u \in C_1$ *with two out-neighbours, update*

$$C_2 := C_2 \cup \{u\}, \quad C_1 := C_1 \setminus \{u\}.$$

Notice that each iteration of Stage 2 decreases the cut-size by at least one.

The Improvement Procedure: Stage 3 (Category (B) vertices)
If the partition is not balanced, the vertices of degree 2 with one out-neighbour must be transferred from the larger part to the smaller part.

If $|C_1| > |C_2|$, relabel $C_1 := C_2$ and $C_2 := V \setminus C_1$.

While $|C_1| < \lfloor \frac{n}{2} \rfloor$ *and there exists a vertex* u *of degree* 2 *in* C_2 *with one neighbour in* C_1*, update*

$$C_1 := C_1 \cup \{u\}, \quad C_2 := C_2 \setminus \{u\}.$$

Each iteration of the while loop in Stage 3 doesn't increase the size of the cut. In the end of Stage 3 if the final partition doesn't have a 2-community structure then a vertex of the category (A) must exist in the partition. In that case, Stage 1 or 2 must be executed before entering Stage 3 again, hence the cut-size is decreased by at least one. Notice that Stage 3 may again create vertices of the category (A) even if they didn't exist before entering Stage 3.

It is easy to see that the algorithm always terminates. Each iteration of the while loop in Stage 1 (resp. Stage 2) decreases the cut-size by at least one. In Stage 3 each iteration of the while loop increases the size of the smaller part by at least one and halts before or when the partition is balanced. Following the construction, if the Improvement Procedure needs to be run again, it must first run through Stage 1 or 2 which decreases the cut-size by at least one. Moreover, the algorithm clearly runs in polynomial time.

Let's discuss the correctness of the algorithm. Suppose the algorithm terminates with the final partition $\{C_1, C_2\}$. Due to the conditions inside the algorithm, $\lceil \frac{n-1}{3} \rceil \leq |C_i| \leq n - \lceil \frac{n-1}{3} \rceil$, $i = 1, 2$.

Initially, the partition is connected and remains so after each stage, hence the final partition is connected too.

Then necessarily, each vertex of degree 1 satisfies the condition (1) since it must be in the same part as its neighbour. Now there are two options:

– If the final partition is balanced then all vertices of degree 2 and 3 may have at most one out-neighbour (otherwise the Improvement Procedure could be applied again), hence the final partition $\{C_1, C_2\}$ is a 2-community structure due to Lemma 3(ii).

– If the final partition is not balanced, then the partition must have the properties described in Lemma 3(i) or (iii) (otherwise, one of Stages 1-2 could be applied again). Hence the final partition $\{C_1, C_2\}$ is a 2-community structure. □

Now we investigate the problem of the existence and finding of a connected 2-community structure in dense graphs. We prove that any graph $G = (V, E)$ of minimum degree $|V| - 3$ has a connected 2-community structure which can be found in polynomial time.

Lemma 5 *If the complement of the graph G is 2-colourable (using each colour for at least 2 vertices), then G has a connected 2-community structure which can be found in polynomial time.*

Proof Let $G = (V, E)$ be a graph such that its complement $\overline{G}$ is 2-colourable. Fix a 2-colouring of $\overline{G}$ (with at least 2 vertices for each colour) and define $\{C_1, C_2\}$ as a partition of V, where each part corresponds to one colour in $\overline{G}$. Obviously, $|C_1|, |C_2| \geq 2$. Notice that the induced subgraph on the vertex set C_1 (resp. C_2) is a clique. Therefore, any vertex $v \in V$ satisfies the condition (1) and the partition $\{C_1, C_2\}$ is a 2-community structure. Since a 2-colouring can be found in polynomial time, the 2-community structure $\{C_1, C_2\}$ too. Obviously, the partition is connected. □

This result directly implies the following theorem:

Theorem 3 *The complement of any bipartite graph (with at least two vertices in each part) has a connected 2-community structure which can be found in polynomial time.*

Theorem 4 *Any graph (except a star) of minimum degree $(n - 3)$, $n \geq 4$, where n is the order of the graph, has a connected 2-community structure which can be found in polynomial time.*

Proof Let G be a graph of size n and of minimum degree $(n - 3)$ (except a star), $n \geq 4$, and $\overline{G}$ be the complement of G. Notice that $\overline{G}$ is of degree at most 2. If $\overline{G}$ doesn't contain an odd cycle, then there exists a 2-colouring of $\overline{G}$ with at least 2 vertices for each colour. In such case, a connected 2-community structure can be found in polynomial time due to Lemma 5.

Now let A be the union of all vertices belonging to an odd cycle in $\overline{G}$ and denote by $B := V \setminus A$. $\overline{G}[A]$ is the union of p odd induced cycles with the vertex sets $O_1, \dots, O_p$, $p \geq 1$. For each i, $1 \leq i \leq p$, let v_i be any vertex of O_i and fix a 2-colouring of $\overline{G}[O_i \setminus \{v_i\}]$. Let $O_{i,1}$, $O_{i,2}$ be the set of vertices corresponding to each colour, obviously $|O_{i,1}| = |O_{i,2}|$. If $|B| \geq 2$, take a 2-colouring of B and define a partition $\{B_1, B_2\}$ of B (each part corresponding to a colour) such that $|B_1| \geq |B_2| \geq 1$, otherwise $B_1 := B$, $B_2 := \emptyset$. Define

$$C_1 := \cup_{i=1}^{p}(O_{i,1} \cup \{v_i\}) \cup B_1, \quad C_2 := \cup_{i=1}^{p} O_{i,2} \cup B_2.$$

Observe that $|C_1|, |C_2| \geq 2$ ($|C_2| \leq 1$ is only possible for a star or a graph with 3 vertices). Obviously, every such 2-colouring can be found in polynomial time. Finally we show that the partition $\{C_1, C_2\}$ is a connected 2-community structure.

All vertices of C_2 satisfy the condition (1) in G since $G[C_2]$ is a clique. For each i, $1 \leq i \leq p$, all neighbours of v_i in $G[C_1]$ satisfy the condition (1) in G since they have all vertices of C_1 as neighbours. Moreover, the non-neighbour of v_i in $G[C_1]$ and v_i itself satisfy the condition (1) in G since $|C_1| > |C_2|$ implies that $\frac{|C_1|-2}{|C_1|-1} \geq \frac{|C_2|-1}{|C_2|}$.

Observe that the partition $\{C_1, C_2\}$ is connected. Obviously, $G[C_2]$ is connected since $G[C_2]$ is a clique. Moreover, any two vertices in C_1 are neighbours except v_i and its neighbour in $\overline{G}[O_{i,1}]$ for all i, $1 \leq i \leq p$. If $B_1 \neq \emptyset$, such two vertices must have a common neighbour in B_1. If $B_1 = \emptyset$, then either $|O_{1,1}| \geq 3$ or $p \geq 2$ (due to assumptions on G), and such two vertices have a common neighbour either in $O_{1,1}$ or $O_{j,1}$, $j \neq i$. Hence, $G[C_1]$ is also connected. □

Theorem 5 *Let $G = (V, E)$ be a graph with minimum degree $\lceil \frac{(c-1).|V|}{c} \rceil$ where c is the size of an inclusion-wise maximal clique in G, i.e. such a clique is not a subgraph of another clique. Then, G has a connected 2-community structure which can be found in polynomial time.*

Proof If $c \geq |V| - 1$, then for any vertex $u \in V$, $d(u) \geq \lceil \frac{(|V|-2).|V|}{|V|-1} \rceil \geq |V| - 3$ and the rest follows from Theorem 4.

If $c \leq |V| - 2$, let C be the inclusion-wise maximal clique in G and take $\{C, V \setminus C\}$ as a partition. Obviously, the size of both parts is at least 2. C is a clique, hence the condition (1) is trivially satisfied for all vertices in C. If a vertex $u \in V \setminus C$ has a neighbour in C, then

$$\frac{d_{in}(u)}{|V| - c - 1} \geq \frac{\frac{(c-1).|V|}{c} - (c-1)}{|V| - c - 1} \geq \frac{c-1}{c} \geq \frac{d_{out}(u)}{c},$$

hence the condition (1) is satisfied for all vertices $u \in V \setminus C$ with a neighbour in C. The rest of vertices in $V \setminus C$ trivially satisfy the condition (1) since they do not have a neighbour in C.

Now we prove that the partition $\{C, V \setminus C\}$ is connected, which is obviously true for $G[C]$. Let suppose that $G[V \setminus C]$ be disconnected and A be the smallest connected component of $G[V \setminus C]$. Notice that $|A| \leq \frac{|V|-c}{2}$ and let $u \in A$. Then $\frac{(c-1)\cdot|V|}{c} \leq d(u) \leq \frac{|V|-c}{2} + c - 2$ and hence $|V| \leq \frac{c\cdot(c-4)}{c-2} < c$, which is impossible. Therefore, $G[V \setminus C]$ is a connected subgraph. □

4 Balanced 2-community structure

In this section we study complexity of the problems related to a balanced 2-community structure. First we prove that every graph of maximum degree 3 has a balanced weak 2-community structure that can be found in polynomial time. The structural properties of low-degree graphs are crucial to obtain such a result. In general graphs, the Balanced Weak 2-community and Balanced 2-community problems are NP-complete as it is shown further in the section. The latter result is contained as the main result in [10], an alternative shorter proof is presented in this section. Both NP-completeness results are extended to a connected balanced 2-community structure.

Remark 1 Due to Theorem 2, every graph of maximum degree 3 has a 2-community structure, but it is not true for a balanced 2-community structure, see Fig. 2. The graph is obtained by linking three "cross gadgets". First notice that if a balanced 2-community exists for the graph, then all vertices of each cross gadget must be in the same part. Indeed, each vertex of such community structure must have two neighbours in its own part. But on the other hand, this graph is impossible to split into two balanced parts without splitting a cross gadget.

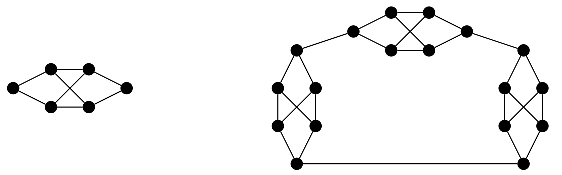

Fig. 2 A cross gadget and a graph of maximum degree 3 without balanced 2-community structure.

Nevertheless, if we focus on a weak community, a balanced weak 2-community always exists in graphs of maximum degree 3, as it is shown in the following theorem.

Theorem 6 *Any graph of maximum degree* 3 *with at least* 4 *vertices has a balanced weak* 2*-community structure. Moreover, such a community structure can be found in polynomial time.*

Proof Let $G = (V, E)$ be a connected graph of maximum degree 3. First notice that in any balanced partition of V:
– each vertex of degree 1 fulfils the condition (2), even if its neighbour is not in its own part,
– each vertex of degree 2 or 3, which has at least one neighbour in its own part, satisfies the condition (2).
Therefore, the only vertices which may not satisfy the condition (2) are vertices of degree 2 or 3 which have no neighbour in their own part.

Choose any balanced partition $\{C_1, C_2\}$ of G and repeat the following steps (S1)-(S2) until it is possible:

(S1) If both parts contain a vertex of degree 2 or 3 that has no neighbour in its own part (say $v_1 \in C_1$, $v_2 \in C_2$), then update: $C_1 := C_1 \cup \{v_2\} \backslash \{v_1\}, C_2 := C_2 \cup \{v_1\} \backslash \{v_2\}$.

(S2) If there is only one partition that contains a vertex v of degree 2 or 3 that has no neighbour in its own part (without loss of generality suppose $v \in C_1$), then choose a vertex $w \in C_2$ such that w has at least one neighbor in C_1 and update: $C_1 := C_1 \cup \{w\} \backslash \{v\}, C_2 := C_2 \cup \{v\} \backslash \{w\}$.

First notice that if case (S2) occurs, such a vertex w always exists since the graph is connected.

Moreover, the partition remains balanced after each step (S1) or (S2). Besides, the cut size between the partitions C_1 and C_2 always decreases (by at least 2 in case (S1), by at least 1 in case (S2)) so after a finite number of iterations (bounded trivially by $O(|V|^2)$, every vertex of degree 2 or 3 has at least one neighbour in its own part. Hence, the algorithm returns a balanced weak 2-community structure. □

Remark 2 Notice that Theorem 6 cannot be extended to a connected case. There exist graphs of maximum degree 3 in which every balanced weak 2-community structures is disconnected, see Fig. 3 as an example.

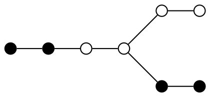

Fig. 3 A tree of maximum degree 3 in which any balanced 2-community structure (or even balanced weak 2-community structure) is disconnected (an example of a balanced 2-community structure is presented by the black and white colours)

Remark 3 It can be observed that the Balanced 2-community problem (hence also Balanced Weak 2-community) is polynomially solvable for graphs with bounded tree-width. Such result follows directly from [3] where the t-Decomposition problem closely related to communities was studied. The input to the t-Decomposition problem is a graph $G = (V, E)$, an integer-valued function $t = t(n)$ such that $0 \leq t(n) \leq n$ for every $n \in \mathbb{N}$, and two functions $a, b : V \to \mathbb{N}$ such that $a(v), b(v) \leq d(v)$, for all $v \in V$. The problem consists of deciding if there is a partition $\{V_1, V_2\}$ of V with $|V_1| = t(|V|)$ such that $d_{G[V_1]}(v) \geq a(v)$ for every $v \in V_1$ and $d_{G[V_2]}(v) \geq b(v)$ for every $v \in V_2$.

In order for $\{V_1, V_2\}$ to be a balanced 2-community structure with $|V_1| \geq |V_2|$, every $v \in V_1$ must satisfy the condition $\frac{d_{G[V_1]}(v)}{\lceil n/2 \rceil - 1} \geq \frac{d(v) - d_{G[V_1]}(v)}{\lfloor n/2 \rfloor}$ and analogously for every $v \in V_2$ must hold $\frac{d_{G[V_2]}(v)}{\lfloor n/2 \rfloor - 1} \geq \frac{d(v) - d_{G[V_2]}(v)}{\lceil n/2 \rceil}$. Thus, Balanced 2-community can be condidered as the t-Decomposition problem for selected values of the functions t, a, b. The conditions for Balanced 2-community can be transformed to the conditions of the t-Decomposition problem where $t(n) = \lceil \frac{n}{2} \rceil$, $a(v) = b(v) = \lceil \frac{n/2-1}{n-1} d(v) \rceil$ for n even and $a(v) = \lceil d(v)/2 \rceil$, $b(v) = \lceil \frac{(n-1)/2-1}{n-1} d(v) \rceil$ for n odd.

Since the t-Decomposition problem was proved to be polynomial-time solvable for bounded tree-width in [3], we can conclude the same result for the Balanced 2-community problem. Notice that the result cannot be extended to a connected case for all graphs, see a tree on Fig. 3 as a counterexample.

Now we focus on the problem of Balanced 2-community in general graphs. In [8] it has been proved that to find a connected balanced partition without any additional constraints is an NP-complete problem in general graphs. We prove similar results for Balanced Weak 2-community and Balanced 2-community and their connected variants. To show that Balanced Weak 2-community is NP-complete, we use a reduction from the Balanced Co-Satisfactory Partition problem, proved to be NP-complete in [5].

The problems is defined as follow:

Balanced Co-Satisfactory Partition
Input : A graph $G = (V, E)$ on an even number of vertices.
Question : Is there a balanced partition $\{C_1, C_2\}$ of V such that for every $v \in V$, $d_{in}(v) \leq d_{out}(v)$?

Theorem 7 Balanced Weak 2-community *is NP-complete.*

Proof The problem is clearly in NP. In the following we define a polynomial-time reduction from Balanced Co-Satisfactory Partition to Balanced Weak 2-community. Let G be a graph on an even number n of vertices as an instance of Balanced Co-Satisfactory Partition, and let $\overline{G}$, the complement of G, be an instance of Balanced Weak 2-community. If G admits a balanced co-satisfactory partition $\{C_1, C_2\}$ then $\{C_1, C_2\}$ is also a weak 2-community. Suppose $d_{in}(v) \leq d_{out}(v)$ for every vertex $v \in V$ (in the graph G). Let $\bar{d}_{in}(v)$ (resp. $\bar{d}_{out}(v)$) be the number of in-neighbours (resp. out-neighbours) of v in $\overline{G}$. Then, the following holds $\bar{d}_{in}(v) + 1 = \frac{n}{2} - d_{in}(v) \geq \frac{n}{2} - d_{out}(v) = \bar{d}_{out}(v)$, which is the condition (2) for a balanced partition. Conversely, any balanced weak 2-community in $\overline{G}$ is a balanced co-satisfactory partition in G. □

The proof of the NP-completeness of Balanced Co-Satisfactory Partition in [5] is based on the graphs $G = (V, E)$, where $V = F \cup T \cup V_0$ with some additional properties: F and T are independent sets, there are no edges between T and V_0, and there is a vertex $f \in F$ that is not adjacent to any vertex of V_0. Any balanced co-satisfactory partition $\{C_1, C_2\}$ of V must have the following structure: $C_1 = F \cup S$ and $C_2 = T \cup (V_0 \setminus S)$ where $S \subseteq V_0$. If $\overline{G}$ is an instance of Balanced Weak 2-community (constructed following the proof of Theorem 7), one can see that C_1 is connected since f is adjacent to all vertices in $F \cup S$ and C_2 is connected since T is a clique and every vertex of T is adjacent to every vertex of $V_0 \setminus S$. Hence we can conclude that even the connected version of Balanced Weak 2-community is NP-complete.

Theorem 8 Connected Balanced Weak 2-community *is NP-complete.*

Estivill-Castro *et al.* [10] have shown that Balanced 2-community is NP-complete by constructing a reduction from a variant of the Clique problem. We propose a shorter alternative proof which is also valid for the Connected Balanced 2-community problem. The proof is based on the NP-complete problem Balanced Satisfactory Partition which was introduced by Bazgan *et al.* [4] as follows:

Balanced Satisfactory Partition
Input : A graph $G = (V, E)$ on an even number of vertices.
Question : Is there a balanced partition $\{C_1, C_2\}$ of V such that for every $v \in V$, $d_{in}(v) \geq \frac{d(v)}{2}$?

It can be proved that these two problems are in fact equivalent when the number of vertices is even.

Lemma 6 *Let $G = (V, E)$ be a graph with n vertices. Consider a partition $\{C_1, C_2\}$ of V and $v \in C_1$. Then the following assertions are equivalent:*

1. $\frac{d_{in}(v)}{|C_1|-1} \geq \frac{d(v)}{n-1}$
2. $\frac{d_{out}(v)}{|C_2|} \leq \frac{d(v)}{n-1}$
3. $\frac{d_{in}(v)}{|C_1|-1} \geq \frac{d_{out}(v)}{|C_2|}$

Proof $(1) \Leftrightarrow (2)$: $\frac{d_{in}(v)}{d(v)} \geq \frac{|C_1|-1}{n-1} \Leftrightarrow 1 - \frac{d_{out}(v)}{d(v)} \geq \frac{n-|C_2|-1}{n-1} \Leftrightarrow 1 - \frac{n-|C_2|-1}{n-1} \geq \frac{d_{out}(v)}{d(v)} \Leftrightarrow \frac{d_{out}(v)}{d(v)} \leq \frac{|C_2|}{n-1}$

$(3) \Leftrightarrow (1)$: $\frac{d_{in}(v)}{|C_1|-1} \geq \frac{d_{out}(v)}{|C_2|} \Leftrightarrow \frac{d_{in}(v)}{|C_1|-1} \geq \frac{d(v)-d_{in}(v)}{n-|C_1|} \Leftrightarrow d_{in}(v)[\frac{1}{|C_1|-1} + \frac{1}{n-|C_1|}] \geq \frac{d(v)}{n-|C_1|} \Leftrightarrow \frac{d_{in}(v)}{d(v)} \geq \frac{|C_1|-1}{n-1}$ □

Note. Notice that the third assertion in Lemma 6 is the condition (1) of a 2-community structure.

Lemma 7 *Let $G = (V, E)$ be a graph with an even number n of vertices and $\{C_1, C_2\}$ be a balanced partition of V. Then for any vertex $v \in V$, $d_{in}(v) = \frac{n/2-1}{n-1} d(v)$ if and only if $d(v) = n - 1$.*

Proof If $d(v) = n - 1$, then clearly $d_{in}(v) = \frac{n}{2} - 1$. Suppose now that $d_{in}(v) = \frac{n/2-1}{n-1} d(v)$. Notice that $(-2)(\frac{n}{2} - 1) + 1(n - 1) = 1$ from which it can be easily shown that $\frac{n}{2} - 1$ and $n - 1$ do not have common divisors. This implies that $d(v)$ is a multiple of $n - 1$. Thus, $d(v) = n - 1$. □

Note. Let $\{C_1, C_2\}$ be a balanced partition of G and $v \in C_1$ be a vertex of degree $n-1$. Since v has $\frac{n}{2}-1$ neighbours in its own part and $\frac{n}{2}$ in other part, v does not satisfy the condition of Balanced Satisfactory Partition. However, v satisfies the Balanced 2-Community condition since $\frac{d_{in}(v)}{|C_1|-1} = 1$.

Proposition 1 *For any graph with n vertices and maximum degree $(n-2)$ the problems* Balanced Satisfactory Partition *and* Balanced 2-Community *are equivalent.*

Proof Suppose that $G = (V, E)$ is a yes-instance of Balanced Satisfactory Partition. Hence there exists a balanced partition $\{C_1, C_2\}$ of V such that any vertex $v \in V$ satisfies the condition $d_{in}(v) \geq \frac{1}{2}d(v)$, which implies that $d_{in}(v) \geq \frac{|C_1|-1}{2|C_1|-1}d(v) = \frac{|C_1|-1}{n-1}d(v)$. Thus, G is a yes-instance of Balanced 2-Community.

Suppose now that G is a yes-instance of Balanced 2-Community. Hence there exists a balanced partition $\{C_1, C_2\}$ of V such that any vertex $v \in V$ satisfies the condition $d_{in}(v) \geq \frac{|C_1|-1}{|C_2|}d_{out}(v)$ that is equivalent to $d_{in}(v) \geq \frac{|C_1|-1}{n-1}d(v)$ using Lemma 6. According to Lemma 7, there is no vertex v such that $d_{in}(v) = \frac{|C_1|-1}{n-1}d(v)$.

Now we need to show that for every vertex $v \in V$, $d_{in}(v) \geq \frac{1}{2}d(v)$. Suppose by contradiction that there exists a vertex $v \in V$ that does not satisfy the inequality that is

$$\frac{|C_1|-1}{n-1}d(v) < d_{in}(v) < \frac{1}{2}d(v)$$

First, notice that $\frac{1}{2}d(v) - \frac{|C_1|-1}{n-1}d(v) = \frac{1}{2(n-1)}d(v) < 1$, which means that there is at most one integer number between $\frac{|C_1|-1}{n-1}d(v)$ and $\frac{1}{2}d(v)$.

Moreover, $d(v)$ cannot be even, since otherwise $\frac{d(v)}{2}$ would be a whole number and thus $d_{in}(v)$ could not be an integer number. Then $d(v)$ is odd and let $d(v) = 2p+1$ for some integer p. We arrive to a contradiction by showing that $p < d_{in}(v) < p + \frac{1}{2}$. Notice that $d(v) < n-1 \Rightarrow \frac{d(v)-1}{2} < \frac{|C_1|-1}{n-1}d(v)$ that implies $p < \frac{|C_1|-1}{n-1}d(v) < d_{in}(v)$. Then necessarily $d_{in}(v) \geq \frac{1}{2}d(v)$ for every vertex $v \in V$, that is G is a yes-instance of Balanced Satisfactory Partition. □

Balanced Satisfactory Partition has already been proved NP-complete in [4], even if both parts are required to be connected. Moreover, the reduction used in [4] does not construct a graph with vertices of degree $n-1$.

Thus we obtain a similar result as in [10] (the authors have mentioned in the proof that used technique works also in a connected case).

Theorem 9 Connected Balanced 2-Community *is NP-complete.*

Finally, it is interesting to notice that there exist graphs in which every 2-community structure is balanced (see Fig. 4).

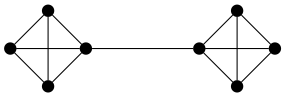

Fig. 4 An example of a graph in which all 2-community structures are balanced

5 Conclusion and open problems

An interesting open question is to determine if a graph of size at least 4 (except stars) has always a 2-community structure, even a connected one. In this paper we prove that the statement is true for trees, graphs of maximum degree 3, minimum degree $|V| - 3$ and some other graph classes. Furthermore, such a structure can be found in polynomial time. The question remains open even for a weak 2-community structure where the partial positive results are only known for the same graph classes.

In case of Balanced 2-Community the situation is different. We show that any graph of maximum degree 3 has a balanced weak 2-community structure, while we present a graph without a balanced 2-community structure within the same class. Computationally speaking, finding a balanced weak 2-community structure can be done in polynomial time in graphs of maximum degree 3 while the Balanced 2-Community problem is NP-complete in general graphs just as its weak version. The results are similar for connected communities.

To get better understanding of community structures, there are some interesting problems left open, as to extend 2-community results to other graph classes, to characterise graph classes where the existential/complexity results for 2-community/weak 2-community problems and their connected versions are different or to generalise the results to k-communities for a fixed k, $k \geq 3$.

References

1. R. Aharoni, E. C. Milner and K. Prikry. Unfriendly partitions of a graph. *Journal of Combinatorial Theory* B, 50:1-10, 1990.
2. K. Andreev and H. Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6):929–939, 2006.
3. C. Bazgan, Z. Tuza, and D. Vanderpooten. Degree-constrained decompositions of graphs: Bounded treewidth and planarity. *Theoretical Computer Science*, 355(3):389–395, 2006.
4. C. Bazgan, Z. Tuza, and D. Vanderpooten. The satisfactory partition problem. *Discrete Applied Mathematics*, 154(8):1236–1245, 2006.
5. C. Bazgan, Z. Tuza, and D. Vanderpooten. Approximation of satisfactory bisection problems. *Journal of Computer and System Sciences*, 74(5):875–883, 2008.
6. C. Bazgan, Z. Tuza, and D. Vanderpooten. Satisfactory graph partition, variants, and generalizations. *European Journal of Operational Research*, 206(2):271–280, 2010.
7. A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz. Recent advances in graph partitioning. *arXiv:1311.3144*.
8. J. Chlebikova. Approximating the maximally balanced connected partition problem in graphs. *Information Processing Letters*, 60(5):223–230, 1996.

9. D. Delling, A. Goldberg, T. Pajor, and R. Werneck. Customizable route planning. In *Proceedings of 10th International Symposium on Experimental Algorithms*, LNCS 6630, pages 376–387, 2011.
10. V. Estivill-Castro and M. Parsa. On connected two communities. In *Proceedings of the 36th Australasian Computer Science Conference (ACSC)*, pages 23–30, 2013.
11. V. Estivill-Castro and M. Parsa. Hardness and tractability of detecting connected communities. In *Proceedings of the Australasian Computer Science Week Multiconference (ACSW)*, Article No. 25, 2016.
12. A. E. Feldmann and L. Foschini. Balanced partitions of trees and applications. *Algorithmica*, 71(2):354–376, 2015.
13. G. Flake, S. Lawrence and C. L. Giles. Efficient identification of web communities. In *Proc. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, pp. 150160, 2000.
14. S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.
15. I. G. Yero and J. A. Rodrguez-Velazquez. *arXiv:1308.2096.*
16. P. Kristiansen, S. M. Hedetniemi and S. T. Hedetniemi. Alliances in graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 48:157-177, 2004.
17. M. E. J. Newman. Detecting community structure in networks. *The Europ. Physical Journal B–Condensed Matter and Complex Systems*, 38(2):321–330, 2004.
18. M. Olsen. A general view on computing communities. *Mathematical Social Sciences*, 66(3):331–336, 2013.
19. S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1:27–64, 2007.
20. K. H. Shafique. Partitioning a graph in alliances and its application to data clustering. *PhD Thesis*, School of Computer Science, University of Central Florida, Orlando, 2004.
21. D. B. Shmoys. Cut problems and their application to divide-and-conquer. In *Approximation Algorithms for NP-Hard Problems, PWS Publishing, 192–235*, 1996.