Theoretical Computer Science 769 (2019) 32–42

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Finding a potential community in networks [☆]

Cristina Bazgan [a,*,1], Thomas Pontoizeau [a], Zsolt Tuza [b,c]

[a] Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016 Paris, France
[b] Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, Hungary
[c] Department of Computer Science and Systems Technology, University of Pannonia, Veszprém, Hungary

ARTICLE INFO

Article history:
Received 12 September 2017
Received in revised form 8 October 2018
Accepted 15 October 2018
Available online 25 October 2018
Communicated by T. Erlebach

Keywords:
Graph
Complexity
Algorithm
Independent set
Inapproximability

ABSTRACT

An independent 2-clique of a graph is a subset of vertices that is an independent set and such that any two vertices inside have a common neighbor outside. In this paper, we study the complexity of finding an independent 2-clique of maximum size in several graph classes and we compare it with the complexity of maximum independent set. We prove that this problem is NP-hard on apex graphs, APX-hard on line graphs, not $n^{1/2-\epsilon}$-approximable on bipartite graphs and not $n^{1-\epsilon}$-approximable on split graphs, while it is polynomial-time solvable on graphs of bounded degree and their complements, graphs of bounded treewidth, planar graphs, (C_3, C_6)-free graphs, threshold graphs, interval graphs and cographs.

1. Introduction

Community detection is a well established research field in the area of social networks [6,12,13,22,25]. It can find many applications in this area with the recent development of social networks like Facebook or Linkedin. A social network can be easily modeled by a graph in which vertices represent members and edges represent relationships between those members.

There are several ways to define a community. Intuitively, a community corresponds to a dense subgraph, that is to say a subgraph with a lot of edges. If a community is defined as a group of maximum size such that all members know each other, it corresponds to the well known NP-hard problem of finding a maximum clique. However, such a condition is strong and is not always relevant to describe a community.

Another way to define a community is to relax the strong condition of a clique and focus on the distance between members of a social network. Different measures have been studied to describe it. Luce introduced in [20] the notion of k-cliques while Mokken extended this notion in [21] by defining k-clubs. A k-clique (resp. a k-club) of G is a subgraph S in which any two vertices are at distance at most k in G (resp. in the subgraph induced by S). The standard term ‘clique’ means both a 1-clique and a 1-club.

With the recent development of social networks and particularly online dating services, it could be interesting to investigate the detection of some group of people who do not know each other, but are related by their other relationships. Such a group could be considered as a ‘potential’ community since it does not form a community in the first place, but could

[☆] A preliminary extended abstract of this paper appeared in Proceedings of the 10th International Conference on Algorithms and Complexity (CIAC 2017), LNCS, vol. 10236, pp. 80–91.

[*] Corresponding author.

E-mail addresses: bazgan@lamsade.dauphine.fr (C. Bazgan), thomas.pontoizeau@lamsade.dauphine.fr (T. Pontoizeau), tuza@dcs.uni-pannon.hu (Z. Tuza).

[1] Institut Universitaire de France.

https://doi.org/10.1016/j.tcs.2018.10.011
0304-3975/© 2018 Elsevier B.V. All rights reserved.

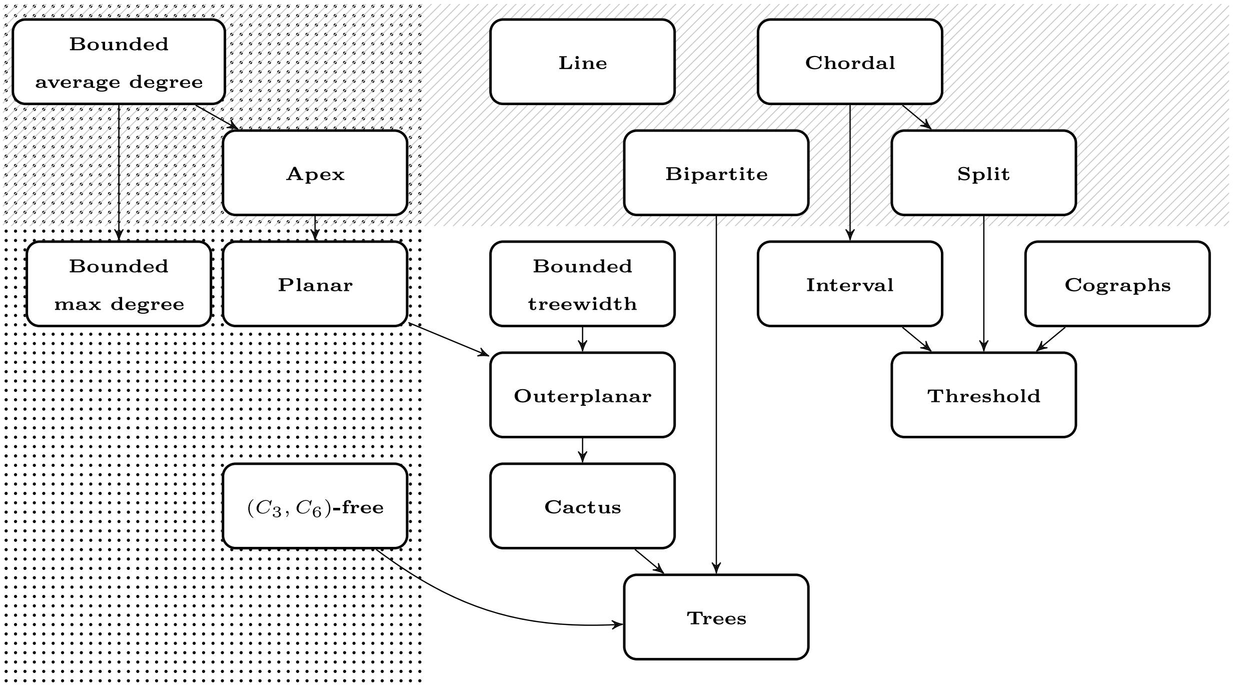

Fig. 1. Relationship among some classes of (connected) graphs, where each child is a subset of its parent. We compare the hardness of Max Independent 2-Clique and Max Independent Set in studied graph classes. Max Independent 2-Clique is NP-hard on graph classes at the top of the figure (hatched area) and is polynomial-time solvable on graph classes at the bottom (non-hatched area). Max Independent Set is NP-hard on graph classes on the left of the figure (dotted area) and is polynomial-time solvable on graph classes on the right (non-dotted area).

become one due to their proximity. This may find various applications in online dating and meet-up services in which members expect not to know the other members.

We propose to define in this paper potential communities as independent k-cliques, for an integer $k \geq 2$. An independent k-clique is an independent set in which any two vertices are at distance at most k in the original graph. Contrary to a k-club, the distance between two vertices must be realized via vertices outside of the subgraph. This definition can have some interesting applications, for instance in automated referee/reviewer seeking. The task would be to identify potential referees/reviewers in an academic community: any two referees should not be directly linked, and at the same time they should not be far away in the network to guarantee the closeness to the content and quality of the reviews. In this paper, we study the problem of finding an independent 2-clique of maximum size.

We investigate the complexity of this problem in several graph classes. Since this problem is close to finding an independent set of maximum size, we also compare the hardness of the two problems. Fig. 1 summarizes the results we prove in the paper.

The paper is structured as follows. In Section 2 we introduce formally some notation and definitions. In Section 3 we show that the complexity of Max Independent 2-Clique jumps from polynomial-time solvable to NP-hard when the input class is extended from planar graphs to apex graphs. In Section 4 we present polynomial algorithms to solve Max Independent 2-Clique in some graph classes. In Section 5 we show NP-hardness and inapproximability of Max Independent 2-Clique in some other graph classes. Conclusions are given in Section 6.

2. Preliminaries

In this paper, all considered graphs are undirected. The *complement* $\overline{G} = (V, \overline{E})$ of a graph $G = (V, E)$ is the graph in which $uv \in E$ if and only if $uv \notin \overline{E}$, for all vertex pairs $u, v \in V$. A *k-cycle* is a cycle of length k. A *block* is a maximal biconnected subgraph. The maximum degree of a vertex in a graph G will be denoted by the usual notation $\Delta(G)$.

We recall that a *clique* in a graph is a set of mutually adjacent vertices. A set of vertices is called a 2-*clique* if any two vertices of the set are at distance at most 2 in G. An *independent set* in a graph is a set of vertices such that no two of them are joined by an edge. An *independent 2-clique* is a subset of vertices which is an independent set and a 2-clique at the same time.

In this paper we are interested in the following optimization problem:

Max Independent 2-Clique
Input: A graph $G = (V, E)$.
Output: A subset $S \subset V$ which is an independent 2-clique of maximum size.

The Max Independent 2-Clique problem is closely related to the well known problem of finding an independent set of maximum size, named Max Independent Set. We sometimes make reductions from the problem Max Clique which refers to the problem of finding a clique of maximum size.

Given a graph G, the standard notation for the maximum size of an independent set in G is $\alpha(G)$. The maximum number of vertices in an independent 2-clique of G will be denoted by $\alpha_{=2}(G)$. The subscript '$=2$' intends to express that the distance between any two vertices of the independent set is exactly 2. Remark that, by definition, $\alpha(G) \geq \alpha_{=2}(G)$.

Note that $\alpha_{=2}(G) \geq 2$ whenever at least one connected component of G is not a complete graph. Indeed, any such component contains two vertices at distance exactly two, hence forming an independent 2-clique of size 2. Moreover, if G is disconnected and has components $G_1, \ldots, G_p$ then

$$\alpha_{=2}(G) = \max_{1 \leq i \leq p} \alpha_{=2}(G_i)$$

For these reasons we assume throughout that G is a non-complete, connected graph (although some of the algorithms also need to handle disconnected graphs temporarily).

We define the problem Independent 2-Clique as the decision version associated to Max Independent 2-Clique. Its input is a graph $G = (V, E)$ and an integer k, and the question is whether there exists an independent 2-clique of size at least k in G.

Some classes of graphs

We study Max Independent 2-Clique in several graph classes. Some definitions are given next. A *cactus* is a graph in which each edge occurs in at most one cycle. A *(C_3, C_6)-free graph* is a graph containing no triangle C_3 and no induced cycle of length 6. A *d-regular* graph is a graph in which all vertices are of degree d. An *interval graph* is a graph for which there exists a family of intervals on the real line and a bijection between the vertices of the graph and the intervals of the family in such a way that two vertices are joined by an edge if and only if the intersection of the two corresponding intervals is non-empty. A graph is a *threshold graph*[2] if it can be constructed from the empty graph by a sequence of two operations: insertion of an isolated vertex, and insertion of a dominating vertex (i.e., a vertex adjacent to all the other vertices). A *cograph* is a graph that can be generated from the single-vertex graph by (repeated applications of) complementation and vertex-disjoint union. A *split graph* is a graph whose vertex set can be partitioned into two subsets, one inducing an independent set S and the other one inducing a clique K. A *bipartite* graph is a graph in which the set of vertices can be partitioned into two independent sets. We denote by $K_{p,m}$ the complete bipartite graph with p and m vertices in its vertex parts. The *line graph* of a graph G is the graph $L(G)$ whose vertices represent the edges of G, and two vertices of $L(G)$ are adjacent if and only if the corresponding two edges of G share a vertex. A connected graph is a *tree* if it does not contain any cycle. A graph is *planar* if it can be embedded in the plane (drawn with points for vertices and continuous curves for edges) without crossing edges. A graph is *outerplanar* if it has a crossing-free embedding in the plane such that all vertices are on the same face. A graph is *k-outerplanar* if for $k = 1$, G is outerplanar and for $k > 1$ the graph has a planar embedding such that if all vertices on the exterior face are deleted, the connected components of the remaining graph are all $(k-1)$-outerplanar. A graph G is *apex* if it contains a vertex v such that $G - v$ is planar. A family of graphs on n vertices is *δ-dense* if it has at least $\frac{\delta n^2}{2}$ edges. It is *everywhere-δ-dense* if the minimum degree is at least δn. A family of graphs is *dense* (resp. *everywhere-dense*) if there is a constant $\delta > 0$ such that all members of this family are δ-dense (resp. everywhere-δ-dense).

Parametrized complexity

A parameterized problem is a subset $Q \subset \Sigma \times \mathbb{N}$ where the first component is a decision problem and the second component is called the parameter of the problem. The class FPT contains every parameterized problem $Q \subset \Sigma \times \mathbb{N}$ for which the question 'Does (x, k) belong to Q?' can be decided by an algorithm that runs in $f(k) \cdot |x|^{O(1)}$ time where $(x, k) \in \Sigma \times \mathbb{N}$ and f is a computable function.

Let $Q_1, Q_2 \subset \Sigma \times \mathbb{N}$ be two parameterized problems. We say that Q_1 FPT-reduces to Q_2 if there exist two computable functions f and g and an algorithm that takes as input an instance $(x_1, k_1) \in \Sigma \times \mathbb{N}$ and outputs a new instance $(x_2, k_2) \in \Sigma \times \mathbb{N}$ in $f(k_1) \cdot |x_1|^{O(1)}$ time such that:

- $(x_1, k_1) \in Q_1 \Leftrightarrow (x_2, k_2) \in Q_2$
- $k_2 \leq g(k_1)$

Downey and Fellows [11] introduced the W-hierarchy as different classes of complexity for parameterized problems. A way to prove that a parameterized problem belongs to $W[t]$ is to construct an FPT-reduction from this problem to a problem known to be in $W[t]$. A parameterized problem is $W[t]$-hard if every problem of $W[t]$ FPT-reduces to it.

[2] The original definition is that the graph admits a vertex labeling with positive real numbers, such that two vertices are adjacent if and only if the sum of their labels is at least (or, at most) a given 'threshold' t.

Approximation, L-reduction, and E-reduction

Given an optimization problem in NPO and an instance I of this problem, we denote by $|I|$ the size of I, by $opt(I)$ the optimum value of I, and by $val(I, S)$ the value of a feasible solution S of instance I. The *performance ratio* of S (or *approximation factor*) is $r(I, S) = \max\left\{\frac{val(I,S)}{opt(I)}, \frac{opt(I)}{val(I,S)}\right\}$. The *error* of S, denoted by $\epsilon(I, S)$, is defined as $\epsilon(I, S) = r(I, S) - 1$.

For a function f, an algorithm is an $f(|I|)$*-approximation*, if for every instance I of the problem, it returns a solution S such that $r(I, S) \leq f(|I|)$.

For proofs concerning APX-hardness, we shall use an approximation-preserving reduction, called L-reduction, which was introduced by Papadimitriou and Yannakakis in [23]. Let A and B be two optimization problems. Then A is said to be L*-reducible* to B if there are two constants $a, b > 0$ such that:

1. there exists a function, computable in polynomial time, which transforms each instance I of A to an instance I' of B such that $opt_B(I') \leq a \cdot opt_A(I)$,
2. there exists a function, computable in polynomial time, which transforms each solution S' of I' to a solution S of I such that $|val(I, S) - opt_A(I)| \leq b \cdot |val(I', S') - opt_B(I')|$.

We recall that a problem is in APX if there exists a polynomial-time approximation algorithm for the problem with an approximation ratio bounded by a constant. A problem is APX-hard if every problem of APX L-reduces to that problem.

The notion of an E-reduction (*error-preserving* reduction) was introduced by Khanna et al. [19]. A problem A is called E*-reducible* to a problem B, if there exist polynomial-time computable functions f and g, and a constant β such that

- f maps an instance I of A to an instance I' of B such that $opt(I)$ and $opt(I')$ are related by a polynomial factor, i.e. there exists a polynomial p such that $opt(I') \leq p(|I|) \cdot opt(I)$,
- g maps any solution S' of I' to a solution S of I such that $\epsilon(I, S) \leq \beta \cdot \epsilon(I', S')$.

An important property of an E-reduction is that it can be applied uniformly to all levels of approximability; that is, if A is E-reducible to B and B belongs to $\mathcal{C}$ then A belongs to $\mathcal{C}$ as well, where $\mathcal{C}$ is a class of optimization problems with any kind of approximation guarantee (see [19]).

Independent 2-Clique belongs to W[1]

From the parametrized complexity point of view, it is interesting to notice the following fact.

Theorem 1. Independent 2-Clique *belongs to* $W[1]$ *on general graphs.*

Proof. We construct an FPT-reduction from Independent 2-Clique to Clique. Let $G = (V, E)$ be an instance of Independent 2-Clique. We construct an instance of Clique by considering the graph $G' = (V, E')$ in which $xy \in E'$ if and only if x and y are exactly at distance 2 in G. It is easy to see that there is an independent 2-clique of size k in G if and only if there is a clique of size k in G'. Since Clique belongs to $W[1]$ (see [11]), Independent 2-Clique also belongs to $W[1]$. □

3. Complexity jump from planar graphs to apex graphs

According to [15], Max Independent Set is known to be NP-hard in planar graphs, and thus also in apex graphs. On the other hand, we prove that Max Independent 2-Clique is polynomial-time solvable on planar graphs but NP-hard on apex graphs. This shows that inserting or removing a single vertex in a graph may dramatically change the complexity of Max Independent 2-Clique.

Theorem 2. Max Independent 2-Clique *is NP-hard on apex graphs.*

Proof. We establish a polynomial reduction from Max Independent Set on cubic planar graphs, which is proved to be NP-hard in [15], to Max Independent 2-Clique on apex graphs. Let $G = (V, E)$ be a cubic planar graph, an instance of Max Independent Set. The instance $G' = (V', E')$ of Max Independent 2-Clique is defined by inserting an additional vertex z that is adjacent to every vertex of V. It is easy to see that $\{z\}$ itself is a one-element non-extendable independent 2-clique, while the independent 2-cliques of G' not containing z are precisely the independent sets of G. □

Theorem 2 implies another interesting result:

Corollary 3. Max Independent 2-Clique *is NP-hard on the class of graphs of average degree at most 5.*

Proof. Cubic graphs on n vertices have $3n/2$ edges, thus the graph constructed in the proof of Theorem 2 is of order $n+1$ and has $5n/2$ edges, yielding average degree less than 5. □

In order to prove that Max Independent 2-Clique is polynomial-time solvable on planar graphs, we use a famous theorem introduced by Courcelle in [9] which states that any problem expressible in Monadic Second-Order Logic is linear-time solvable for graphs of bounded treewidth. This allows to show first the following:

Theorem 4. Max Independent 2-Clique *is linear-time solvable on graphs with bounded treewidth.*

Proof. We observe that Max Independent 2-Clique is expressible in Monadic Second-Order Logic:

$$maxI2C(S) := max_S\{|S| : \forall x \forall y (Sx \wedge Sy) \rightarrow (\neg edg(x,y) \wedge (\exists z, edg(x,z) \wedge edg(y,z)))\}$$

Since any problem expressible in Monadic Second-Order Logic is linear-time solvable for graphs of bounded treewidth (see [9]), $\alpha_{=2}$ can be determined in linear time in graphs of bounded treewidth. □

Based on this result, we prove the following.

Theorem 5. Max Independent 2-Clique *is polynomial-time solvable on planar graphs.*

Proof. Let $G = (V, E)$ be a planar graph and $v \in V$ any vertex. Then all the other vertices in an independent 2-clique S containing v are at distance exactly 2 apart from v. Further, the 2-clique property for $S \setminus \{v\}$ is ensured by vertices within distance at most 3 from v. Thus, the vertices relevant for S to be an independent 2-clique induce a subgraph G' in G such that G' belongs to the class of '4-outerplanar' graphs. Graphs which are 4-outerplanar have treewidth at most 11 (more generally, all k-outerplanar graphs have treewidth at most $3k-1$, due to [4]). Then, using Theorem 4, a polynomial-time algorithm for Max Independent 2-Clique in planar graphs consists in solving the problem for all subgraphs G' (which have treewidth at most 11) defined from each vertex v of G and choosing a solution of maximum size. □

In order to get the following result we can use the elementary Five Color Theorem for planar graphs, and even Euler's formula without any further topological considerations, but we prefer to cite in our proof the Four Color Theorem because it leads to a better kernel.

Theorem 6. *The parameterized problem associated with* Max Independent 2-Clique *with the natural parameter is in FPT on apex graphs.*

Proof. Let $G = (V, E)$ be an apex graph and $x \in V$ a vertex such that $G - x$ is planar. Since any planar graph is 4-colorable [3], the size of an independent set in $G[N(x)]$ is at least $\frac{|N(x)|}{4}$, and so is the size of an independent 2-clique in G. Thus, considering the parameter k, if $|N(x)| \geq 4k$, then the answer is yes.

If now $|N(x)| < 4k$, let S be any independent 2-clique and $v \in S$ any vertex. As discussed in the previous proof, the 2-clique property for $S \setminus \{v\}$ is ensured by vertices within distance at most 3 from v in $G - x$, and by vertices in the closed neighborhood $N[x]$ of x. For this reason, for each vertex $v \in V$, we consider the subgraph induced by the set of all vertices at distance at most 3 from v and include $\{x\} \cup N(x)$. This subgraph has treewidth at most $12 + |N(x)| < 12 + 4k$. Since any problem parameterized by q expressible in Monadic Second-Order Logic is in FPT with respect to q on graphs of treewidth bounded by q [10], a polynomial-time algorithm can be designed by solving the problem for all such subgraphs defined from each vertex v of G, and answering yes if at least one such problem answers yes and answering no otherwise. □

4. Graph classes with polynomial-time algorithms

In the following we identify some graph classes on which Max Independent 2-Clique is computable in polynomial time, while Max Independent Set is not always polynomial-time solvable.

First, it is interesting to notice that, according to the next propositions, Max Independent 2-Clique is polynomial-time solvable on graphs of bounded degree and also on complements of graphs of bounded degree, while Max Independent Set is NP-hard on graphs of bounded degree [15] but polynomial-time solvable on their complements (using exhaustive search in the non-neighborhood of each vertex, which can be done in linear time).

Proposition 7. Max Independent 2-Clique *is linear-time solvable on graphs with bounded maximum degree.*

Proof. The proof consists in computing, for each vertex v of a graph $G = (V, E)$, the largest size of an independent 2-clique v can belong to. Since the maximum degree is bounded, also the number of vertices at distance 2 from v is bounded, thus the largest independent 2-clique among them can be determined in constant time. Performing this for all vertices of the graph can be done in $O(|V|)$ steps. □

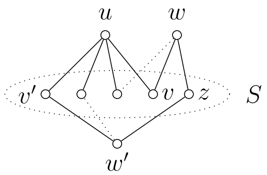

Fig. 2. The independent 2-clique S and its (partial) neighborhood selected in the proof of Lemma 10. Dotted lines are possible edges, so z can be at distance 2 from other vertices in S but those are unimportant for the proof.

Proposition 8. Max Independent 2-Clique *is linear-time solvable on graphs of minimum degree at least* $(n-d)$*, where* d *is constant.*

Proof. Since every vertex is non-adjacent with fewer than d vertices, the size of a solution cannot exceed d. Then using an exhaustive search in the non-neighborhood of each vertex, we can find an optimal solution in linear time. □

Now, notice that a natural way to find an independent 2-clique is to take an independent set included in the neighborhood of one vertex. First, this principle can be applied easily on trees.

Proposition 9. *Every tree* T *satisfies* $\alpha_{=2}(T) = \Delta(T)$*. Thus,* Max Independent 2-Clique *is linear-time solvable on trees without using Monadic Logic.*

Proof. Any two vertices v, w of an independent 2-clique S share a neighbor, say u, which is unique in any tree. Non-neighbors of u cannot belong to S because they are at distance at least 3 from v or w (or both). On the other hand, all neighbors of u have mutual distance 2, so that $|S|$ is largest if S is the neighborhood of a vertex of maximum degree. □

In this way, it is interesting to investigate the properties of a graph in which an independent 2-clique is not included in the neighborhood of one vertex. We show in Lemma 10 that such a graph necessarily contains a cycle of length 3 or 6, and cannot be a cactus if such an independent 2-clique has a certain size. Such properties allow us to get an easy polynomial-time algorithm for Max Independent 2-Clique on (C_3, C_6)-free graphs, while Max Independent Set is NP-hard[3] on this class of graphs (see [1]). From Theorem 4 we already know that Max Independent 2-Clique is linear-time solvable on cactus graphs, but the property of Lemma 10 allows us to give a simpler algorithm for this class of graphs.

Lemma 10. *Let* $G = (V, E)$ *be a graph. Suppose that there exists an independent 2-clique* S *not contained in the neighborhood of a single vertex. Then* G *contains an induced cycle of length* 3 *or* 6*. Moreover, if* $|S| \geq 4$*, then* G *is not a cactus.*

Proof. Let S be an independent 2-clique in G such that not all vertices of S have a common neighbor. Let u be a vertex in $V \setminus S$ which has the maximum number of neighbors in S, and N_u be the neighborhood of u in S. Then there exists a vertex z in S which is not a neighbor of u. Let v be any vertex of N_u, and w be a common neighbor of z and v. Let v' be a vertex in N_u non-adjacent to w (it exists by the choice of u). Since S is a 2-clique, v' and z have a common neighbor, say w' (notice that w' can be neither u nor w). Thus, $C := (u, v, w, z, w', v', u)$ is a cycle in G (see Fig. 2).

If C has no chord, then it is an induced 6-cycle of G; and otherwise any chord of C lies inside $\{u, w, w'\}$ and thus it creates a 3-cycle in G. This proves the first assertion.

Suppose now that $|S| \geq 4$. Then there are three options:

- u has only two neighbors in S. Then any two vertices of S must have a different common neighbor in $V \setminus S$ (by the choice of u), moreover there exists z' in $S \setminus \{N_u, z\}$. In this situation v, z, z' with their three pairwise neighbors create a 6-cycle sharing the edge wz with C and thus G is not a cactus.
- u has at least 3 neighbors and w has only v as a neighbor in N_u. Let z' be a vertex of $N_u \setminus \{v', v\}$. Then z and z' must have a common neighbor x (which cannot be u or w but could be w'). Then wz is a common edge of C and the 6-cycle (u, z', x, z, w, v, u) and thus G is not a cactus.
- u has at least 3 neighbors and w has at least 2 neighbors in N_u, say v and z'. Then vw is a common edge of C and the 4-cycle (u, v, w, z', u) and thus G is not a cactus. □

This lemma implies the following theorem:

Theorem 11. *Any* (C_3, C_6)*-free graph* G *satisfies* $\alpha_{=2}(G) = \Delta(G)$ *and* Max Independent 2-Clique *is linear-time solvable on it.*

[3] It is proved in [1] that for a finite set H of connected graphs, Max Independent Set is NP-hard on the class of H-free graphs if no member of H is either a path or a tree with one vertex of degree 3 and the other vertices of degree at most 2.

Proof. By Lemma 10, in (C_3, C_6)-free graphs any independent 2-clique is the neighborhood of some vertex. Then, an independent 2-clique of maximum size is given in linear time by taking the neighborhood of a vertex of maximum degree since vertices in the neighborhood of any vertex are not adjacent in C_3-free graphs. □

Finally, Lemma 10 allows us to give a polynomial-time algorithm for Max Independent 2-Clique on cactus graphs.

Proposition 12. Max Independent 2-Clique *is linear-time solvable on cactus graphs.*

Proof. Since all cycles of a cactus can be triangulated without creating a K_4, cactus graphs have treewidth at most 2, hence an implicit algorithm running in linear time follows from the proof of Theorem 4.

Being more constructive, Lemma 10 implies that an independent 2-clique S in a cactus of order at least two either is a single vertex of a C_3, or consists of at most three (independent) vertices of a C_6, or lies entirely in the neighborhood of a vertex v. In the latter case, if $B_1, \dots, B_k$ are the blocks incident with v, then if B_i is an edge or a triangle then it can have just one vertex in S; and if B_i is a longer cycle then both neighbors of v in B_i can belong to S. Since every set obtained in this way is an independent 2-clique, the maximum size can be determined in linear time. □

We focus in the following part of this section on classes of graphs on which both Max Independent 2-Clique and Max Independent Set are polynomial-time solvable. We first investigate a subclass of split graphs, namely threshold graphs. It follows from the definitions that a threshold graph $G = (V, E)$ is a split graph with the following property: the vertices of the independent set S can be ordered as $v_1, \dots, v_p$ such that $N_G(v_1) \subseteq N_G(v_2) \subseteq \dots \subseteq N_G(v_p)$. We denote by $u_1, \dots, u_q$ the vertices of the clique K, and we suppose that $d_G(u_1) \leq d_G(u_2) \leq \dots \leq d_G(u_q)$. Without loss of generality, we assume that there is no isolated vertex in G. Note that a threshold graph can be recognized in linear time (see [18]).

Proposition 13. Max Independent 2-Clique *is linear-time solvable on threshold graphs. Moreover, in every threshold graph G without isolated vertices we have $\alpha_{=2}(G) = \alpha(G)$.*

Proof. Let $G = (V, E)$ be a threshold graph with the previous decomposition into S and K. Let $N_G(v_p) = \{u_r, u_{r+1}, \dots, u_q\}$, for some $r \geq 1$. Then a maximum independent 2-clique in G is S if $K \setminus N_G(v_p) = \emptyset$, and otherwise it is $S \cup \{z\}$ with any $z \in K \setminus N_G(v_p)$, since in both cases the common neighbor of all these vertices is u_q. Since Max Independent Set can be solved in linear time in threshold graphs [14], Max Independent 2-Clique can also be solved in linear time. □

The previous result can be extended in two directions, for interval graphs and for cographs.

Using the results of Booth and Lueker [5] it can be tested in linear time whether a graph G is an interval graph; and if it is, then an interval representation $I_1, \dots, I_n$ of G can also be generated.

Proposition 14. Max Independent 2-Clique *is polynomial-time solvable on interval graphs.*

Proof. Consider any $G = (V, E)$ and let $I_1, \dots, I_n$ be an interval representation of G. In order to determine $\alpha_{=2}(G)$, first notice that all vertices of an independent 2-clique S of G must have a common neighbor. Indeed, if I and I' are the leftmost and the rightmost intervals of S then any of their common neighbors intersects all intervals located between them, and therefore is a common neighbor of all members of S. Then, for every vertex I, we compute a maximum independent set in the subgraph induced by the neighborhood of I. An optimal solution is such an independent set with maximum size. Since Max Independent Set is polynomial-time solvable on interval graphs [16], the result follows. □

We consider now the class of cographs, that contains all threshold graphs, and we show that Max Independent 2-Clique is also polynomial-time solvable on this class. To each cograph G with n vertices, we can associate a rooted tree T, called the *cotree* of G. Leaves of T correspond to vertices of the graph G, and internal nodes of T are labeled with either '$\cup$' (union-node) or '$\times$' (join-node). A subtree rooted at node '$\cup$' corresponds to the vertex-disjoint union of the subgraphs defined by the children of that node, and a subtree rooted at node '$\times$' corresponds to the complete join of the subgraphs defined by the children of that node; that is, we add an edge between every two vertices corresponding to leaves in different subtrees under the join-node in question. Cographs can be recognized in linear time and the cotree representation can be obtained efficiently [8,17]. Moreover, any cotree can easily be transformed in linear time to a binary cotree with $O(n)$ nodes.

Proposition 15. Max Independent 2-Clique *is polynomial-time solvable on cographs.*

Proof. Consider a cograph G with n vertices $v_1, \dots, v_n$. Given a binary cotree representation T of G with $O(n)$ nodes, we show in the following how to solve Max Independent 2-Clique recursively.

Let $x_1, \dots, x_t$ be the nodes of T where t is in $O(n)$. For $i = 1, \dots, t$, denote by T_i the subtree rooted at x_i, G_i the subgraph induced by the vertices corresponding to the leaves of T_i, and V_i the set of these vertices.

For each i, we compute $\alpha_{=2}(G_i)$ 'bottom-up' in the cotree. We start by computing values of leaves, and after that the value of an internal node if the values of its two children are already computed. Together with $\alpha_{=2}(G_i)$ we also determine the independence number $\alpha(G_i)$, which is well known to admit an easy recursion (which follows immediately by the constructive definition of cographs).

Given a node x_i of the cotree, the corresponding values are obtained as follows:

- If x_i is a leaf then $\alpha_{=2}(G_i) = |V_i| = 1$. Also, $\alpha(G_i) = 1$.
- If x_i is a union-node with two children x_ℓ and x_r, we have no edges between G_ℓ and G_r. Then any maximum independent 2-clique of G_i is entirely contained either in G_ℓ or in G_r. So, $\alpha_{=2}(G_i) = \max\{\alpha_{=2}(G_\ell), \alpha_{=2}(G_r)\}$. On the other hand, clearly, $\alpha(G_i) = \alpha(G_\ell) + \alpha(G_r)$.
- If x_i is a join-node with two children x_ℓ and x_r, every vertex in V_ℓ is adjacent to every vertex in V_r. Then a maximum independent 2-clique in G_i is a maximum independent set entirely contained either in G_ℓ or in G_r. So, $\alpha_{=2}(G_i) = \alpha(G_i) = \max\{\alpha(G_\ell), \alpha(G_r)\}$.

Since each step can be performed in constant time, moreover postorder traversal requires linear time, the algorithm runs proportionally to the size of the cotree, which is $O(n)$. □

Notice that since Max Independent Set is linear-time solvable on chordal graphs [14], it is also polynomial-time solvable on interval graphs and threshold graphs. Moreover, Max Independent Set is also polynomial-time solvable on cographs by bottom-up tree computation [7].

5. NP-hardness and inapproximability

We investigate graph classes in which Max Independent 2-Clique is NP-hard and, in some cases, inapproximable. Using first the reduction from the proof of Theorem 2, we can conclude:

- Max Independent 2-Clique is NP-hard on dense (resp. everywhere dense) graphs, since Max Independent Set is NP-hard on dense (resp. everywhere dense) graphs. Moreover, Max Independent 2-Clique is not $n^{1-\varepsilon}$-approximable for any $\varepsilon > 0$, if P $\neq$ NP, on everywhere dense graphs (and respectively dense graphs) since the same result holds for Max Independent Set on everywhere dense graphs (and respectively dense graphs). In order to get this last result, we use the same inapproximability result for Max Independent Set on general graphs [26] and a reduction preserving approximation from general graphs to everywhere dense graphs (that consists of adding a clique of the same size as the size of the graph and joining every vertex from the original graph to all vertices in this clique).
- Max Independent 2-Clique is NP-hard on K_4-free graphs, since Max Independent Set is NP-hard on K_3-free graphs [1].

We now investigate graph classes in which Max Independent 2-Clique is NP-hard while Max Independent Set is polynomial-time solvable. We first consider a graph class containing threshold graphs, namely the class of split graphs, for which Max Independent 2-Clique becomes NP-hard (and even not $n^{1-\varepsilon}$-approximable, unless P = NP). Since Max Independent Set is polynomial-time solvable on chordal graphs [14], it is also polynomial-time solvable on split graphs.

Proposition 16. Max Independent 2-Clique *is NP-hard on split graphs.*

Proof. We reduce Max Clique on general graphs to Max Independent 2-Clique on split graphs. Let $G = (V, E)$ be an instance of Max Clique. We define an instance $G' = (V', E')$ of Max Independent 2-Clique on split graphs as follows: for every vertex $v_i \in V$ we consider a vertex $v'_i \in V'$ and for every edge $e \in E$ we consider a vertex e' in V'. We also add an additional vertex z in V'. Moreover, for any edge $e = v_1 v_2 \in E$ we associate two edges in E', the edges $v'_1 e'$ and $v'_2 e'$. Finally, the subgraph induced by vertices $e' \in V'$ and z is defined to be a clique. Now it is easy to see that C is a clique of size at least k in G if and only if $C' = \{v' : v \in C\} \cup \{z\}$ is an independent 2-clique of size at least $k+1$ in G'. On the other hand, given an independent 2-clique S, if $z \notin S$ and $e' \in S$ holds for some (only one) $e \in E$, then we can modify S to an independent 2-clique of the same size by replacing e' with z. Hence, the maximum can always be attained by involving z. □

Theorem 17. Independent 2-Clique *is W[1]-complete on split graphs.*

Proof. From Theorem 1, we know that Independent 2-Clique belongs to W[1]. On the other hand, the reduction in Proposition 16 is an FPT-reduction. Since Clique is W[1]-hard on general graphs [11], it follows that Independent 2-Clique is also W[1]-hard on split graphs. □

Theorem 18. Max Independent 2-Clique *is not $n^{1-\varepsilon}$-approximable in polynomial time on split graphs unless $P = NP$.*

Proof. We construct an E-reduction from Max Clique. Let $I = (V, E)$ be an instance of Max Clique and let $I' = (V', E')$ be the corresponding instance of Max Independent 2-Clique, considering the same reduction as in Proposition 16. First,

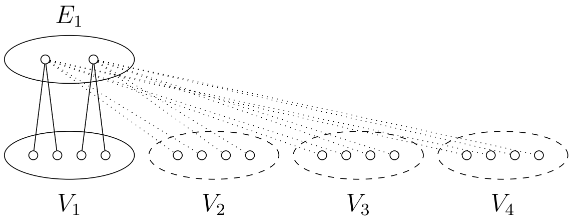

Fig. 3. The bipartite graph G', an instance of Max Independent 2-Clique.

notice that $opt(I) = opt(I') - 1$, thus we have $opt(I') \leq 2\,opt(I)$. Now let S' be an independent 2-clique of I' of size at least 2 and let S be the set of all copies of vertices from V in S'. Since $opt(I) = opt(I') - 1$ and $|S| = |S'| - 1$, we obtain $opt(I) - |S| = opt(I') - |S'|$. Since it has been proved in [26] that Max Clique is not $n^{1-\varepsilon}$-approximable in polynomial time unless P = NP, Max Independent 2-Clique is not $n^{1-\varepsilon}$-approximable in polynomial time on split graphs unless P = NP. □

We prove now that Max Independent 2-Clique is NP-hard (and even not $n^{1/2-\epsilon}$-approximable, unless P = NP) on bipartite graphs while Max Independent Set is polynomial-time solvable since the number of vertices in a maximum independent set equals the number of edges in a minimum edge covering.

Proposition 19. Max Independent 2-Clique *is NP-hard on bipartite graphs.*

Proof. Max Independent Set is known to be NP-hard on 3-regular graphs [15], so Max Clique is also NP-hard on $(n-4)$-regular graphs (where n is the number of vertices), by considering its complement. We reduce Max Clique on $(n-4)$-regular graphs to Max Independent 2-Clique on bipartite graphs. Let $G = (V, E)$ be an $(n-4)$-regular graph. We construct an instance $G' = (V', E')$ of Max Independent 2-Clique on bipartite graphs as follows (see Fig. 3).

Let V_1, V_2, V_3, V_4 be four copies of V. Let E_1 be a set of $|E|$ vertices corresponding to the edges in E, and define $V' := V_1 \cup V_2 \cup V_3 \cup V_4 \cup E_1$. Let there exist an edge in E' between a vertex v in V_i, $i \in \{1, 2, 3, 4\}$ and a vertex e in E_1 if and only if the corresponding vertex v in V is incident with the corresponding edge e in E.

Now we show that G contains a clique of size at least k if and only if G' contains an independent 2-clique of size at least $4k$.

Given a clique $C \subseteq V$ of size at least k in G, the union of the four copies of C in G' is an independent 2-clique of size at least $4k$.

For the other direction, notice first that the value of a maximum independent set in a 3-regular graph is at least $\lceil \frac{n}{4} \rceil$. Then, the value of a maximum clique in an $(n-4)$-regular graph is also at least $\lceil \frac{n}{4} \rceil$. Thus the size of a maximum independent 2-clique in G' is at least n.

We consider now a solution C' of Max Independent 2-Clique in G' with at least $4k \geq n$ vertices (this restriction is always possible because of the previous comment). Notice that C' cannot contain both a vertex from E_1 and a vertex from $V' \setminus E_1$ since the distance between any two vertices of C' must be 2. A solution which is a subset of E_1 would mean pairwise intersecting edges in G, hence would have size at most $\max(3, n-4) < n$. Therefore C' must be a subset of $V' \setminus E_1$. Notice that for any $i \in \{1, 2, 3, 4\}$, $C' \cap V_i$ must be a copy of a clique in G. Then C' is a union of copies of four cliques in G, and $|C'| \geq 4k$. Let C_0 be the copy of largest size, which thus has $|C_0| \geq k$. Then C_0 is the copy of a clique C of G of size at least k. □

Theorem 20. Max Independent 2-Clique *is not $n^{1/2-\varepsilon}$-approximable in polynomial time on bipartite graphs, unless $P = NP$.*

Proof. We construct an E-reduction from Max Clique. Let $I = (V, E)$ be an instance of Max Clique. Consider a reduction similar to the one in the proof of Proposition 19, except that we now consider $\ell = |V|$ copies $V_1, \ldots, V_\ell$ instead of four copies of V; adjacencies are defined in the same way as before. We denote by $I' = (V', E')$ the corresponding instance of Max Independent 2-Clique from the reduction. As in Proposition 19, starting with a clique of size $opt(I)$, we can construct an independent 2-clique of size $\ell \cdot opt(I)$ in G' and thus $opt(I') \geq \ell \cdot opt(I)$. Let S' be any independent 2-clique in I' of size at least ℓ (it always exists, take e.g. the ℓ copies of the same vertex, one copy in each V_i). As before, S' cannot contain both a vertex of E_1 and a vertex from $V \setminus E_1$ since two vertices of S' must have distance 2 in G', and S' cannot contain only vertices from E_1 since any independent 2-clique included in E_1 is of size at most $\max(3, \Delta(G)) \leq \ell - 1$. Moreover, each subset $V_i \cap S'$ corresponds to a clique in G. Let S be the subset $V_i \cap S'$ of largest size. We have $|S| \geq \frac{|S'|}{\ell}$ and then $opt(I) \geq |S| \geq \frac{|S'|}{\ell} = \frac{opt(I')}{\ell}$ when S' is an optimal solution. Using that $opt(I') \geq \ell \cdot opt(I)$ we get $opt(I') = \ell \cdot opt(I)$ and we obtain:

$$\epsilon(I, S) = \frac{opt(I)}{|S|} - 1 \leq \frac{\ell \cdot opt(I')}{\ell \cdot |S'|} - 1 = \epsilon(I', S')$$

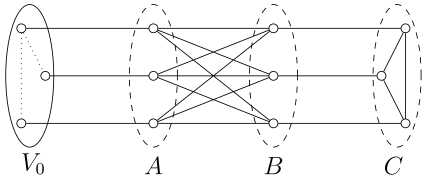

Fig. 4. The graph G' for which the corresponding line graph $L(G')$ is an instance of Max Independent 2-Clique.

Since we clearly have $opt(I') \leq p(|I|) \cdot opt(I)$ with a polynomial p, the reduction is an E-reduction. Then, since Max Clique is not $\ell^{1-\varepsilon}$-approximable, unless P = NP [26], the same property holds for Max Independent 2-Clique. Thus Max Independent 2-Clique is not $n^{1/2-\varepsilon}$ approximable where $n = |V'|$ since $n = \ell^2 + |E|$. □

Finally we prove that Max Independent 2-Clique is NP-hard (and even APX-hard) on line graphs, while Max Independent Set is polynomial-time solvable since it consists in a maximum matching in the original graph.

Proposition 21. Max Independent 2-Clique *is NP-hard on line graphs.*

Proof. We establish a reduction from the Max Clique problem on general graphs. Consider an instance $G = (V, E)$ of Max Clique with $|V| = n$. We construct a graph $G' = (V', E')$ (see Fig. 4) as follows. Let $G_0 = (V_0, E_0)$ be a copy of G. Let V' be $V_0 \cup A \cup B \cup C$ where A, B, C are three sets of n vertices. Then, let $E' = E_0 \cup E_1 \cup E_2 \cup E_3 \cup E_4$ such that E_1 is a perfect matching between V_0 and A, E_2 is the set of all possible edges (i.e., a complete bipartite graph) between the vertices of A and the vertices of B, E_3 is a perfect matching between B and C, and E_4 is the set of all possible edges between any two vertices of C (a complete subgraph). The line graph of G', denoted by $L(G')$, is an instance of Max Independent 2-Clique. Notice that an independent 2-clique in $L(G')$ corresponds to a set of edges in G' such that, for each pair of edges $\{e_1, e_2\}$ in the set, e_1 and e_2 are not adjacent but are joined by an edge. We show that G contains a clique of size at least k if and only if $L(G')$ contains an independent 2-clique of size at least $k + n$.

Consider a clique S of size k in G, and let S_0 be its copy in G'. We define a set of edges S' of size at least $k + n$ in G' as follows. For any vertex $v \in S_0$, add in S' its adjacent edge in E_1. Moreover add the entire E_3 to S'. We show now that any pair of edges in S' have an adjacent edge in common. Two edges of $S' \cap E_1$ have a common adjacent edge in E_0 since the subgraph induced by S_0 is a clique. Similarly, two edges of E_3 have a common adjacent edge in E_4. Moreover, an edge of $S' \cap E_1$ and an edge of E_3 have a common adjacent edge in E_2 since the subgraph induced by $A \cup B$ is $K_{n,n}$. Then, the corresponding set of vertices in $L(G')$ is an independent 2-clique of size $k + n$.

In the other direction, consider an independent 2-clique in $L(G')$ of size $k + n$. Notice that it is always possible to take the set of vertices in $L(G')$ corresponding to E_3 in G' and two edges in E_1 whose vertices in V_0 are neighbors in G', hence we can suppose that $k \geq 2$. Let S' be the set of all corresponding edges in G'. Suppose first that there is exactly one edge from E_0 in S'. Then, there are at most $n - 2$ edges from E_1 in S', and there are at most 2 edges from E_2 in S', due to the constraints of an independent 2-clique. There cannot be edges from $E_3 \cup E_4$ in S' since they would not be joined to the edge of $E_0 \cap S'$ by any edge. Then, S' contains at most $n + 1$ edges in S', which contradicts $k \geq 2$. Suppose now that there are $g \geq 2$ edges from E_0 in S'. Those edges of E_1 which meet them, are excluded from S', hence there are at most $n - 2g$ edges from E_1 in S'; and there is no edge from E_2 in S' because the distance-2 neighbors of each $e \in E_2$ form a star in E_0, while the current $S' \cap E_0$ is a matching of g edges. Then the size of S' is at most $n - g$, which contradicts $k \geq 2$.

Thus, we can assume that there is no edge from E_0 in S'. Similarly, there is no edge from E_4 in S'. Now, notice that $|S' \cap (E_2 \cup E_3)| \leq n$ since if $S' \cap (E_2 \cup E_3)$ contained $n + 1$ edges then at least two of these edges would have a common endpoint. Consequently, $|S' \cap E_1| \geq k$. Moreover, any two edges from $S' \cap E_1$ must have a common adjacent edge in E_0 since they cannot have a common adjacent edge in E_2. Then, the subgraph of G induced by the set of vertices in V_0 which are the endpoints of the edges in $S' \cap E_1$ must be a clique whose size is at least k. □

Theorem 22. Max Independent 2-Clique *is APX-hard on line graphs.*

Proof. We construct now an L-reduction from a restricted version of Max Clique to Max Independent 2-Clique on line graphs. Let I be an instance of Max Clique on graphs of degree at least $n - 4$ and I' the corresponding instance of Max Independent 2-Clique on line graphs from the previous reduction. We prove that this reduction is an L-reduction. We proved in Proposition 21 that any independent 2-clique in I' has a size at most $2n$. Then $opt(I') \leq 2n = 8 \cdot \frac{n}{4} \leq 8 \cdot opt(I)$ follows since $opt(I) \geq \frac{n}{4}$ in graphs of degree at least $n - 4$. Moreover, starting with a clique of size $opt(I)$, we can construct an independent 2-clique of size $opt(I) + n$ and therefore $opt(I') \geq n + opt(I)$. Let S' be an independent 2-clique in I' of size at least $n + 2$ (we proved in Proposition 21 that it always exists and that such a set must be included in $E_1 \cup E_2 \cup E_3$). Let S be the set of vertices in V_0 which are incident with edges in $E_1 \cap S'$. We have $|S'| - |S| \leq n$, i.e. $n + |S| \geq |S'|$. Then we obtain $opt(I) - |S| \leq opt(I') - n - |S| = opt(I') - (n + |S|) \leq opt(I') - |S'|$. Since Max Independent Set is APX-hard on the class of graphs of maximum degree 3 [2], Max Clique is also APX-hard on the class of graphs of minimum degree at least $n - 4$. Thus, Max Independent 2-Clique is APX-hard on line graphs. □

6. Conclusion

Despite that Max Independent 2-Clique and Max Independent Set are similar problems, their complexity can be very different depending on the graph class we try to solve the problem in. We showed that Max Independent 2-Clique is NP-hard on apex, dense and everywhere dense, K_4-free, split, bipartite and line graphs while it is polynomial-time solvable on bounded treewidth, planar, bounded degree (and complement of bounded degree), (C_3, C_6)-free, interval graphs and cographs. Many further types of graphs may be of interest, concerning separation of graph classes in which the problem is NP-hard from the ones where the problem is solvable in polynomial time.

From the parameterized complexity point of view, we showed that Independent 2-Clique is in FPT on apex graphs. On the other hand, some problems related to the parameterized complexity are open. Independent Set is in FPT on K_3-free graphs [24] but Independent 2-Clique remains open for this class of graphs. From Proposition 19, Clique on $(n - 4)$-regular graphs FPT reduces to Independent 2-Clique in bipartite graphs but Clique is in FPT on $(n - 4)$-regular graphs (since Independent Set is in FPT on 3-regular graphs). Then the parameterized complexity of Independent 2-Clique remains open on bipartite graphs.

Acknowledgement

Research of Zsolt Tuza was supported in part by the National Research, Development and Innovation Office – NKFIH under the grant SNN 116095.

References

[1] V.E. Alekseev, On the local restrictions effect on the complexity of finding the graph independence number, in: Combinatorial–Algebraic Methods in Applied Mathematics, 1983, pp. 3–13.
[2] P. Alimonti, V. Kann, Some APX-completeness results for cubic graphs, Theoret. Comput. Sci. 237 (1–2) (2000) 123–134.
[3] K. Appel, W. Haken, Every planar map is four colorable: Part I. Discharging, Illinois J. Math. 21 (1977) 429–490.
[4] H.L. Bodlaender, A partial k-arboretum of graphs with bounded treewidth, Theoret. Comput. Sci. 209 (1998) 1–45.
[5] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, J. Comput. System Sci. 13 (3) (1976) 335–379.
[6] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, C. Schulz, Recent advances in graph partitioning, in: Algorithm Engineering – Selected Results and Surveys, in: LNCS, vol. 9220, Springer, 2016, pp. 117–158.
[7] D.G. Corneil, H. Lerchs, L.S. Burlingham, Complement reducible graphs, Discrete Appl. Math. 3 (3) (1981) 163–174.
[8] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (4) (1985) 926–934.
[9] B. Courcelle, The monadic second-order logic of graphs III: tree-decompositions, minors and complexity issues, RAIRO Inform. Théor. Appl. 26 (1992) 257–286.
[10] B. Courcelle, J. Engelfriet, Graph structure and monadic second-order logic: a language-theoretic approach, vol. 138, Cambridge University Press, 2012.
[11] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness II: on completeness for W[1], Theoret. Comput. Sci. 141 (1) (1995) 109–131.
[12] D. Easley, J. Kleinberg, Networks, Crowds, and Markets: Reasoning about a Highly Connected World, Cambridge University Press, 2010.
[13] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3–5) (2010) 75–174.
[14] A. Frank, Some polynomial algorithms for certain graphs and hypergraphs, Congr. Numer. XV (1976) 3–13.
[15] M.R. Garey, D.S. Johnson, L. Stockmeyer, Some simplified NP-complete problems, Theoret. Comput. Sci. 1 (3) (1976) 237–267.
[16] U.I. Gupta, D.T. Lee, J.Y.-T. Leung, Efficient algorithms for interval graphs and circular-arc graphs, Networks 12 (4) (1982) 459–467.
[17] M. Habib, C. Paul, A simple linear time algorithm for cograph recognition, Discrete Appl. Math. 145 (2) (2005) 183–197.
[18] P. Heggernes, D. Kratsch, Linear-time certifying recognition algorithms and forbidden induced subgraphs, Nordic J. Comput. 14 (2007) 87–108.
[19] S. Khanna, R. Motwani, M. Sudan, U.V. Vazirani, On syntactic versus computational views of approximability, SIAM J. Comput. 28 (1) (1998) 164–191.
[20] R.D. Luce, Connectivity and generalized cliques in sociometric group structure, Psychometrika 15 (1950) 169–190.
[21] R.J. Mokken, Cliques, clubs and clans, Qual. Quant. 13 (2) (1979) 161–173.
[22] M.E.J. Newman, Detecting community structure in networks, Eur. Phys. J. B 38 (2) (2004) 321–330.
[23] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, J. Comput. System Sci. 43 (3) (1991) 425–440.
[24] V. Raman, S. Saurabh, Triangles, 4-cycles and parameterized (in-)tractability, in: Algorithm Theory – SWAT 2006: 10th Scandinavian Workshop on Algorithm Theory, in: LNCS, vol. 4059, 2006, pp. 304–315.
[25] S.E. Schaeffer, Graph clustering, Comput. Sci. Rev. 1 (1) (2007) 27–64.
[26] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, Theory Comput. 3 (1) (2007) 103–128.