J Comb Optim (2012) 23:94–117
DOI 10.1007/s10878-010-9343-5

The MAX QUASI-INDEPENDENT SET problem

N. Bourgeois · A. Giannakos · G. Lucarelli · I. Milis · V.T. Paschos · O. Pottié

Published online: 14 August 2010

Abstract In this paper, we deal with the problem of finding quasi-independent sets in graphs. This problem is formally defined in three versions, which are shown to be polynomially equivalent. The one that looks most general, namely, f-MAX QUASI-INDEPENDENT SET, consists of, given a graph and a non-decreasing function f, finding a maximum size subset Q of the vertices of the graph, such that the number of edges in the induced subgraph is less than or equal to $f(|Q|)$. For this problem, we show an exact solution method that runs within time $O^*(2^{\frac{d-27/23}{d+1}n})$ on graphs of average degree bounded by d. For the most specifically defined γ-MAX QUASI-

This work is partially supported by the French Agency for Research under the DEFIS program TODO, ANR-09-EMER-010. An extended abstract of this paper has been included in the proceedings of the CSR'10 conference, LNCS 6072.

N. Bourgeois · A. Giannakos · G. Lucarelli (✉) · V.T. Paschos · O. Pottié
LAMSADE, CNRS FRE 3234 and Université Paris-Dauphine, Paris, France
e-mail: lucarelli@lamsade.dauphine.fr

N. Bourgeois
e-mail: bourgeois@lamsade.dauphine.fr

A. Giannakos
e-mail: giannako@lamsade.dauphine.fr

V.T. Paschos
e-mail: paschos@lamsade.dauphine.fr

O. Pottié
e-mail: pottie@lamsade.dauphine.fr

G. Lucarelli · I. Milis
Department of Informatics, Athens University of Economics and Business, Athens, Greece

G. Lucarelli
e-mail: gluc@aueb.gr

I. Milis
e-mail: milis@aueb.gr

INDEPENDENT SET and k-MAX QUASI-INDEPENDENT SET problems, several results on complexity and approximation are shown, and greedy algorithms are proposed, analyzed and tested.

Keywords Quasi independent set · Exact algorithms · Approximation algorithms

1 Introduction and preliminaries

The problem of finding in a graph a maximum size subgraph whose density differs (being smaller or larger) from that of the whole graph, arises often in various application contexts. For example, inputs may represent graphs, wherein dense (with respect to the input) subgraphs are sought, as it is the case for call details database mining (Abello et al. 2002), or protein-protein interaction networks analysis (Hartwell et al. 1999). In other cases, inputs may represent graphs from which one wants to extract maximum size, sparser than the input graphs, as for example in visualization tools for stock market interaction graphs (Boginski et al. 2005).

In this paper we address the problem of finding in a graph a maximum size subgraph whose sparsity is less than or equal to a value specified with the input. In the case that appears as most general, the sparsity of a graph is measured by means of a function bounding the number of edges in the sought subgraph and depending on its size; we also study some special forms of this function, namely, when it has the form of the ratio of the number of edges of the solution to the number of edges in a complete graph of equal size, and also when it is a numeric parameter of the input.

We denote by G a simple finite graph without loops, by V and $E(G)$ its vertex set and its edge set, respectively, and by n and m their respective sizes. Let A, B be two subsets of V. The induced subgraph by A in G is denoted by $G[A]$ and its edge set by $E[A]$, respectively. The edge set with one extremity in A and the other in $B \setminus A$ will be denoted as $E[A, B]$. Clearly, if A and B are disjoint then $E[A, B] = E[B, A]$ and $E[A, A] = \emptyset$. The *degree of* A *towards* B is equal to $|E[A, B]|$ and is denoted by $\delta_B(A)$; when A is reduced to a singleton $\{v\}$, we denote its degree in B by $\delta_B(v)$, or simply by $\delta(v)$ whenever $B = V$. The maximum vertex degree in a graph G is denoted by Δ_G or simply by Δ if there is no risk of confusion. We also set $d_B(A) = 1/|A| \sum_{v \in A} \delta_B(v)$, and $d = 1/n \sum_{v \in V} \delta(v)$.

We tackle the following variants of the quasi-independent set problem.

f-MAX QUASI-INDEPENDENT SET (the general quasi-independent set problem) Given a graph G and a polynomially computable non-decreasing real function $f : \mathbb{N} \to \mathbb{R}$, find the largest possible $Q \subseteq V$ such that $|E[Q]| \leq f(|Q|)$.

In the above definition, f is used as a *sparsity specification* for the induced subgraph of the sought solution. We study in particular two variants of f-MAX QUASI-INDEPENDENT SET, denoted by γ-MAX QUASI-INDEPENDENT SET and k-MAX QUASI-INDEPENDENT SET, respectively, formally defined in what follows.

In the first one, sparsity specification is given in the special form of the ratio of the number of edges in the subgraph induced by the quasi-independent set over the number of edges induced by a complete graph of the same size:

γ-MAX QUASI-INDEPENDENT SET Given a graph G and a real γ, $0 \le \gamma \le 1$, find the largest possible $Q \subseteq V$ such that $|E[Q]| \le \gamma\binom{|Q|}{2}$.

It is easy to see that γ-MAX QUASI-INDEPENDENT SET is not hereditary (a problem is said hereditary if its solutions satisfy some non-trivial hereditary property[1]). Indeed, given a feasible solution for γ-MAX QUASI-INDEPENDENT SET, the induced subgraph $G[Q']$ of a subset Q' of Q may violate the sparsity condition $|E(Q')| \leqslant \gamma|Q'|(|Q'|-1)/2$.

In the second restricted variant of the problem considered in the paper, we simply seek for a maximum vertex subset of the graph with no more than a constant number of edges having both extremities in it:

k-MAX QUASI-INDEPENDENT SET Given a graph G and a positive integer k, find the largest possible $Q \subseteq V$ such that $|E[Q]| \le k$.

Clearly, k-MAX QUASI-INDEPENDENT SET is hereditary. In fact, it is easy to see that k-MAX QUASI-INDEPENDENT SET belongs to the family of node-deletion problems, first defined in Krishnamoorthy and Deo (1979) and further studied in Yannakakis and Lewis (1980). Formally, a node-deletion problem consists of finding, given a graph G and a non-trivial hereditary property P, the minimum number of vertices of G that one has to delete from G, in order to have P satisfied in the remaining graph. In Yannakakis and Lewis (1980), it is proved that the decision version of such problems is **NP**-complete even for planar graphs.

For $f = 0$ (resp., $\gamma = 0$ and $k = 0$), Q is simply a maximum independent set in G, while for $f : f(|Q|) \ge m$ (resp., for $\gamma \ge 2m/n(n-1)$ and $k \ge m$), $Q = V$ is a trivial solution; being a direct generalization of MAX INDEPENDENT SET, the f-, γ- and k-MAX QUASI-INDEPENDENT SET problems are obviously inapproximable within better than $O(n^{1-\epsilon})$, unless $\mathbf{P} = \mathbf{NP}$ (Zuckerman 2006).

In Abello et al. (2002) essentially the same problem as γ-MAX QUASI-INDEPENDENT SET is addressed, formulated as the research, given a graph and $0 \le \gamma \le 1$, of a maximum subgraph of sparsity (defined as the ratio of the number of its edges over the number of edges of the complete graph of the same size) at least γ; any solution to this problem can be obtained as the complementary of a quasi-independent set of sparsity at most $1-\gamma$ in the complement of the input graph. The authors present an algorithm equivalent to the greedy algorithm for γ-MAX QUASI-INDEPENDENT SET, analyzed later in this paper; however, they focus in implementation issues on very large instances and they don't attempt to analyze its performance. In Jagota et al. (2001) an alternative generalization of independent sets, namely k-insulated sets, is given. These are subsets of the vertex set of the graph such that any vertex in a k-insulated set is adjacent to at most k vertices in it, while any external vertex is adjacent to at least $k+1$ vertices of the k-insulated set.

To our knowledge, the k-MAX QUASI-INDEPENDENT SET problem has been specifically formulated for the first time in Hochbaum and Goldschmidt (1997). The authors call it the *k-edge-in subgraph* problem. Nevertheless, in this paper the prob-

[1] A graph G is said to satisfy a hereditary property π if every subgraph of G satisfies π whenever G satisfies π. Furthermore, π is non-trivial if it is satisfied for infinitely many graphs and it is false for infinitely many graphs; for instance, properties "independent set", "clique", "planar graph", "k-colorable graph", etc., are non-trivial hereditary properties.

lem is not addressed, but simply mentioned as related to other subgraph problems on which it focuses.

A kind of dual of the maximum quasi-independent set problem is to search, given a graph and a positive integer k, for the sparsest—or densest—(maximal) subgraph with exactly k vertices. This kind of problems have been extensively studied during the last years under the names of "k-sparsest" or "k-densest subgraph" problem (see, for example, Asahiro et al. 2000; Feige et al. 2001; Hochbaum and Goldschmidt 1997). Similar problems have been studied in previous works in Goldberg (1984), Picard and Queyranne (1982). However, these papers define sparsity as the ratio of a positive power of the size of a graph over the number of its induced edges.

The remainder of the paper is organized as follows. Section 2 gives several bounds to the optimal solutions for γ-MAX QUASI-INDEPENDENT SET. Section 3 tackles a specific polynomial case for the three variants of MAX QUASI-INDEPENDENT SET and proves their **NP**-hardness in bipartite graphs. Interestingly enough, the three variants of MAX QUASI-INDEPENDENT SET handled in the paper are shown to be polynomially equivalent (with respect to their exact solution) though only one of them is hereditary. In Sect. 4 an exact solution method with non-trivial worst-case running time for the general f-MAX QUASI-INDEPENDENT SET problem is presented and analyzed. As we discuss here this method applies also to other combinatorial optimization problems. Finally, in Sect. 5 approximation results are proved for both γ-MAX QUASI-INDEPENDENT SET (Subsect. 5.1) and k-MAX QUASI-INDEPENDENT SET (Subsect. 5.2).

In what follows, when we indifferently refer to either one of the quasi-independent set versions defined above, we use the term MAX QUASI-INPENDENT SET instead.

2 Solution properties and bounds

As it is already mentioned, in general, γ-MAX QUASI-INDEPENDENT SET is not a hereditary problem; just consider an instance where the graph is an edge plus some isolated vertices, and γ is the smallest possible for having the whole graph as a trivial solution. Obviously, the sparsity condition will be violated for any strict part of the solution containing the edge. However, γ-MAX QUASI-INDEPENDENT SET is still a weakly hereditary problem, in the sense given by the following lemma that will be used later.

Lemma 1 *Let Q be a γ-quasi-independent set in G, of size $q \geq 2$. Then, for any $k \leq q$, there exists in G some γ-quasi-independent set $R(k) \subseteq Q$, of size k.*

Proof Let v be a vertex in $\mathbf{argmax}_{u\in Q}\{\delta_Q(u)\}$. By removing v from Q we get $Q' = Q \setminus \{v\}$ of size $q-1$ where

$$\begin{aligned} |E(Q')| &= |E(Q)| - \delta_Q(v) \leq |E[Q]| - \frac{2|E[Q]|}{|Q|} \\ &\leq \frac{\gamma|Q|(|Q|-1) - 2\gamma(|Q|-1)}{2} = \frac{\gamma(|Q|-1)(|Q|-2)}{2} \end{aligned}$$

Therefore, Q' is a γ-quasi-independent set in G of size $q-1$. □

Next lemma gives some bounds for the solutions of the γ-MAX QUASI-INDEPENDENT SET.

Lemma 2 *Let Q be any non-trivial γ-quasi-independent set* ($0 < \gamma < m/\binom{n}{2}$), *with size q, not contained in any γ-quasi-independent set of size $q+1$. Consider $\vartheta(Q) = \min_{v \in V \setminus Q}\{\delta_Q(v)\}$ and let $Q^* \neq Q$ be an optimal solution for γ-*MAX QUASI-INDEPENDENT SET, *q^* be its size, and for any vertex-subset P, let $d(P)$ be the average degree of the subgraph induced by P* (*recall that we denote $d(V)$ by d*). *Finally, let $\alpha_{\min}$ be the size of a smallest maximal independent set* (*minimum independent dominating set*) *in G. Then*: (1) $q^* \leq \alpha_{\min}(\Delta+1)$; (2) $\frac{q^*}{q} \leq \frac{\Delta}{\vartheta(Q)}$; (3) $q \geq n - \frac{\Delta}{\gamma}$; (4) $q^* \leq \frac{\Delta}{\gamma}$; (5) $q^* \leq \sqrt{\frac{dn}{\gamma}}$; (6) $q^* \leq \frac{d(Q^*)+2}{\gamma} - 1$.

Proof Let S be a smallest maximal independent set in G, i.e. $\alpha_{\min} = |S|$, denote by Q^* a quasi-independent set of maximum size, and set $S' = Q^* \cap S$. By the maximality of S, $q^* - |S'| \leq \delta_S(Q^* \setminus S') \leq |S|\Delta$; hence $q^* \leq |S|\Delta + |S'| \leq \alpha_{\min}(\Delta+1)$ and item (1) is proved.

We now prove item (2). By the definition of Q, it is $\vartheta(Q) > 0$ and:

$$\vartheta(Q)(n-q) \leq \delta_V(Q) = \sum_{v \in Q} \delta(v) - 2|E[Q]| \leq \Delta q \tag{1}$$

By a similar argument, it holds that $\vartheta(Q)(q^* - |Q^* \cap Q|) \leq \delta_{Q \setminus (Q^* \cap Q)}(Q^* \setminus (Q^* \cap Q)) \leq \Delta(q - |Q^* \cap Q|)$, and hence:

$$\frac{q^*}{q} \leq \frac{q^* - |Q^* \cap Q|}{q - |Q^* \cap Q|} \leq \frac{\Delta}{\vartheta(Q)} \tag{2}$$

By the definition of Q, we have also that:

$$\gamma\frac{q(q+1)}{2} < |E[Q]| + \vartheta(Q) \leq \gamma\frac{q(q-1)}{2} + \vartheta(Q) \Rightarrow \vartheta(Q) > \gamma q \tag{3}$$

Combining (1) and (3) we get $\gamma q(n-q) \leq \Delta q \Rightarrow q \geq n - \frac{\Delta}{\gamma}$, that proves item (3), while combining (2) and (3) we get immediately $q^* \leq \frac{\Delta}{\gamma}$, proving so item (4).

Assuming that Q^* is non-trivial, we get:

$$\frac{\gamma q(q+1)}{2} < m = \frac{dn}{2} \Rightarrow \frac{\gamma q^2}{2} < \frac{dn}{2} \Leftrightarrow q^* \leq \sqrt{\frac{dn}{\gamma}}$$

that proves item (5), and also that:

$$\frac{\gamma q(q+1)}{2} < |E[Q \cup v]| \leq |E[Q]| + q = \frac{d(Q)q}{2} + q$$
$$\Leftrightarrow \quad \gamma(q+1) < d(Q) + 2 \Rightarrow q^* < \frac{d(Q^*)+2}{\gamma} - 1$$

that proves item (6) and completes the proof of the lemma. □

By item (4) of Lemma 2 the following easy corollary holds.

Corollary 1 *If γ is bounded below by some positive constant, then γ-*MAX QUASI-INDEPENDENT SET *is polynomial for graphs with bounded degree.*

3 Complexity results for max quasi-independent problems in various graph-classes

3.1 Relations between f-, γ- and k-MAX QUASI-INDEPENDENT SET problems

The following proposition claims that all three variants of MAX QUASI-INPENDENT SET dealt in this paper are closely interrelated.

Proposition 1 *f-, γ- and k-*MAX QUASI-INDEPENDENT SET *are polynomially equivalent with respect to their exact solution.*

Proof (i) We first show that γ-MAX QUASI-INDEPENDENT SET $\preceq$ f-MAX QUASI-INDEPENDENT SET. This is trivial since the condition for sparsity formed as expression of γ is, as noted before, just one particular case of a sparsity specification defined by $f(q) = \gamma q(q-1)/2$.

(ii) We now show that f-MAX QUASI-INDEPENDENT SET $\preceq$ k-MAX QUASI-INDEPENDENT SET: indeed, consider an instance $I = (G, f)$ of the f-MAX QUASI-INDEPENDENT SET problem, and let $Q^*(I)$ be any of its optimal solutions; denote its size by $q^*(I)$. Let $|E[Q^*(I)]| = k' \leq f(q^*(I))$ be the number of the edges in this optimal; clearly, $Q^*(I)$ should be also an optimal solution for the instance (G, k') of the k-MAX QUASI-INDEPENDENT SET problem, i.e., some instance (G, i) with $0 \leq i \leq m$. Thus, $q^*(I)$ has to be the size of one of the optimal solutions $Q^*(G, i)$ of the family of instances $\{(G, i) : 0 \leq i \leq m\}$, and more specifically of such an optimal $Q^*(G, i)$ that satisfies $\mathbf{argmax}_i\{q^*(G, i) : |E[Q^*(G, i)]| \leq f(q^*(G, i))\}$. Since there are at most m of k-MAX QUASI-INDEPENDENT SET instances to solve, the claim follows.

(iii) Now it remains to show that k-MAX QUASI-INDEPENDENT SET$\preceq$ γ-MAX QUASI-INDEPENDENT SET. In a similar manner as before, consider some instance $I = (G, k)$ of the k-MAX QUASI-INDEPENDENT SET, and let $Q^*(I)$ be any of its optimal solutions; denote its size by $q^*(I)$. Let $|E[Q^*(I)]| = k' \leq k$ be the number of edges in this optimal; clearly, $Q^*(I)$ must also be an optimal solution for some instance $(G, \gamma(q^*(I), k'))$ with $\gamma(q^*(I), k') = \frac{2k'}{q^*(I)(q^*(I)-1)}$ of the γ-MAX QUASI-INDEPENDENT SET problem; notice that $0 \leq k' \leq k \leq m$ and, w.l.o.g., $2 \leq q^*(I) \leq n$. Thus, $q^*(I)$ has to be the size of one of the optimal solutions $Q^*(G, \gamma(i, j))$ of the family of instances of γ-MAX QUASI-INDEPENDENT SET, $\{(G, \gamma(i, j)) : \gamma(i, j) = \frac{2i}{j(j-1)}, 0 \leq i \leq k, 2 \leq j \leq n\}$, and more specifically of such an optimal $Q^*(G, \gamma(i, j))$ that satisfies $\mathbf{argmax}_{0\leq i\leq m, 2\leq j\leq n}\{q^*(G, \gamma(i, j)) : |E[Q^*(G, \gamma(i, j))]| \leq k\}$. Since there are at most nm instances of the γ-MAX QUASI-INDEPENDENT SET to solve, the claim follows. □

3.2 Bipartite graphs

We now tackle MAX QUASI-INDEPENDENT SET in bipartite graphs. The following result characterizes its complexity.

Theorem 1 MAX QUASI-INDEPENDENT SET *is* ***NP****-hard on bipartite graphs.*

Proof We first prove that k-MAX QUASI-INDEPENDENT SET is **NP**-hard. Our reduction goes from the r-SPARSEST SUBGRAPH problem, which consists of seeking, given a graph $G(V, E)$ and an integer $1 < r < n$, a subset $H \subset V$ of size r such that $G[H]$ has a minimum number of edges, among all induced subgraphs of size r. In the decision version of r-SPARSEST SUBGRAPH, we are given $G(V, E)$ and two positive integers r and l and we ask if there a subset $H \subset V$ of size r such that $G[H]$ has *at most* l edges. The r-SPARSEST SUBGRAPH problem is **NP**-complete for bipartite graphs by a reduction from the r-DENSEST SUBGRAPH problem. The decision version of this later problem asks for a subset $H \subset V$ of size r such that $G[H]$ has *at least* l edges, and it is known to be **NP**-complete for bipartite graphs (Corneil and Perl 1984).

Consider the decision version of k-MAX QUASI-INDEPENDENT SET: given G and two positive integers t and k, is there $Q \subseteq V$ such that $|Q| \geq t$ and $G[Q]$ has at most k edges? It is straightforward to see that given an instance $I = (G, r, l)$ of (the decision version of) r-SPARSEST SUBGRAPH, the instance $I' = (G, t = r, k = l)$ of k-MAX QUASI-INDEPENDENT SET is a "yes"-instance if I is a "yes"-instance too. On the other hand, if I' has a solution, say Q^* with $|Q^*| > r$, one can easily get a solution for I, just by eliminating any $|Q^*| - r$ vertices from Q^* (recall that k-MAX QUASI-INDEPENDENT SET is hereditary).

Putting together Proposition 1 and the above result, the **NP**-hardness of γ- and f-MAX QUASI-INDEPENDENT SET is directly derived. □

There are several hereditary graph classes the definitions of which imply direct conditions on their sparsity, independently of the measure used; take for instance complete graphs or split graphs. Such proprieties, together with heredity, can be exploited in order to polynomially solve the k-MAX QUASI-INDEPENDENT SET (in fact, by Proposition 1, any of the three variants of MAX QUASI-INDEPENDENT SET). In the sequel, we present a polynomial algorithm for k-MAX QUASI-INDEPENDENT SET, that works on split graphs.

3.3 Split graphs

Let $S = (I, C, E)$ be a split graph, where I is an independent set, C is a clique, and E is the set of edges between I and C plus the edges of the clique C. The following lemma holds.

Lemma 3 *There is an optimal k-quasi-independent set on a split graph* $S = (I, C, E)$ *such that it contains the independent set* I.

Proof Let $I' \cup C'$, $I' \subset I$, $C' \subseteq C$, be the set of vertices selected by an optimal solution. If we remove a vertex $c \in C'$ from this optimal we remove at least $|C'| - 1$ edges. If we add a vertex $i \in I \setminus I'$ we add at most $|C'| - 1$ edges. Hence, the number of edges in $S[(C' \setminus \{c\}) \cup (I' \cup \{i\})]$ is reduced and the new solution is again feasible and optimal. Thus, we can create an optimal solution that includes the set I. □

Based upon Lemma 3, the following theorem holds:

Theorem 2 MAX QUASI-INDEPENDENT SET *is polynomial on split graphs.*

The optimal solution Q^* to k-MAX QUASI-INDEPENDENT SET on a split graph $S = (I, C, E)$ can be found in polynomial time. Indeed, by Lemma 3, Q^* can be initialized to I. Next, we consider the vertices of the clique C in increasing order with respect to their degree, that is in increasing order with respect to the number of edges between any vertex $c \in C$ to its neighbors in I. Using this order, we add vertices to Q^* until the number of edges of $S[Q^*]$ becomes greater than k. The proof for this greedy selection is straightforward.

4 Exact solution of MAX QUASI-INDEPENDENT SET problem

In this section we give an exact algorithm for f-MAX QUASI-INDEPENDENT SET with non-trivial worst-case running time. Let us note that to our knowledge, no algorithm that optimally solves MAX QUASI-INDEPENDENT SET with running time better than $O^*(2^n)$ is known, where in $O^*(\cdot)$ notation polynomial terms are ignored. Also, as we will see in the sequel, the scope of the results of this section is even larger than the MAX QUASI-INDEPENDENT SET case. Indeed, the method described in what follows concerns a broad class of optimization problems, those that "match vertex branching", defined in Definition 1, below.

4.1 Problems that match vertex branching

The intuition behind the exact solution method for MAX QUASI-INDEPENDENT SET, lies in the possibility of organizing the solution space of the problem in a tree-like manner. So we need first to formally characterize the class of optimization graph-problems for which such an organization is possible. This is done in Definition 1.

Definition 1 We say that a graph problem Π matches vertex branching, if for any graph instance $G(V, E)$, for any $v \in V$, there exist some sets of parameters $\mathcal{S}_i$, some subsets $v \in H_i \subset V$ and two functions f_1, f_2 bounded above by some polynomial of n, such that:

$$\mathrm{opt}_\Pi(G, \mathcal{S}_3) \leq \max\left\{f_1(\mathrm{opt}_\Pi(G[V \setminus H_1], \mathcal{S}_1)), f_2(\mathrm{opt}_\Pi(G[V \setminus H_2], \mathcal{S}_2))\right\}$$

where $\mathrm{opt}_\Pi(G, \mathcal{S})$ denotes the value of the optimal solution of Π for G with parameter set $\mathcal{S}$.

Notice that, with appropriate choice for f_1, f_2, it is possible to replace max by min, or to make a single reduction.

Several problems that aim at finding a specific subset in a given graph may be generalized as problems that match vertex branching. For example, for the MAXIMUM WEIGHTED INDEPENDENT SET: Given a graph $G(V, E)$ and a weight function $w : V \to \mathbb{R}$, we search for an independent set S maximizing $\sum_{v \in S} w(v)$, we have $\text{opt}(G, w) \leq \max\{\text{opt}(G[V \setminus v], w), \text{opt}(G[V \setminus N[v]], w) + w(v)\}$. Obviously, this remains true for the non-weighted version, i.e., whenever $w = 1$.

Also the f-MAX QUASI-INDEPENDENT SET can be reformulated as a problem that matches vertex branching, in the following manner: Given a graph $G(V, E)$, two constants w_0, q_0 and a weight function $w : V \to \mathbb{R}$, we search for a maximal size vertex subset $Q \subseteq V$ whose induced graph $G[Q] = (Q, R)$ verifies $|R| + w_0 + \sum_{v \in Q} w(v) \leq f(|Q| + q_0)$. Let $w_+ \equiv w + 1$ on $N(v) = \{u : \{u, v\} \in E\}$ and $w_+ \equiv w$ elsewhere. Then, it is $\text{opt}(G, w_0, q_0, w) \leq \max\{\text{opt}(G[V \setminus v], w_0, q_0, w), 1 + \text{opt}(G[V \setminus v], w_0 + w(v), q_0 + 1, w_+)\}$, and the formulation is completed by setting initially $q_0 = w_0 = 0$, $w \equiv 0$.

Informally, w_0 and q_0 stand, respectively, for the number of edges and of vertices that are already in the solution, while $w(v)$ represents the number of edges that will be added, if one decides to keep v.

Notice that, as it can be shown by straightforward recurrence, any problem that matches vertex branching can be solved within time $O^*(2^n)$. This time-bound already holds for problems where feasible solutions are vertices inducing subgraphs of the input graph satisfying some specific property, since they can be solved in $O^*(2^n)$.

However, an exact algorithm for f-MAX QUASI-INDEPENDENT SET based upon vertex branching would be interesting if its running time $T(n)$ could be shown to be in $2^{\phi(\Delta)n}$ with ϕ some increasing function bounded above by 1 for any Δ. Intuitively, a possibility for such an improvement lies on finding an efficient vertex branching rule for fast reduction of the remaining graph's degree, and showing fast (polynomial) algorithms for solving a maximum f-MAX QUASI-INDEPENDENT SET problem in bounded degree graphs.

4.2 Bottom-up algorithms

4.2.1 General scheme

We give below a general scheme, using vertex branching for finding a maximum f-quasi-independent set in a graph G. Recall that by Proposition 1 such a method can be used for computing an optimal solution for any of the three variants of the MAX QUASI-INDEPENDENT SET problem, with a polynomial overhead. This scheme, parameterized by a graph G, some integer function f, two integers q_0 and w_0, and some vertex weight function w, can be written as follows:

procedure exactrec($G(V, E)$, f, q_0, w_0, w)
in, not_in: integer;
    if $(V = \emptyset)$ **then**
        if $(w_0 \leq f(q_0))$ **then**
            return q_0

```
        else
          return −∞
        endif;
      endif;
      pick v ∈ V(G) such that δ_V(v) = max;
      not_in ← exactrec(G[V \ v], f, q0, w0, w);
      for all u neighbors of v
          w(u) ← w(u) + 1
      endfor;
      in ← exactrec(G[V \ v], f, q0 + 1, w0 + w(v), w);
      return max{in, not_in};
end exactrec;
procedure f_QIS(G(V, E): graph, f: integer function)
      return exactrec(G, f, 0, 0, 0);
end f_QIS
```

As noted at the end of the previous subsection, the running time for an exact method based upon the above scheme can be improved if the f-MAX QUASI-INDEPENDENT SET problem is shown polynomial on graphs of bounded small degree.

Lemma 4 *Assume that some problem that matches vertex branching can be computed on graphs whose average degree is at most* $d-1$, $d \in \mathbb{N}$, *within time* $O^*(2^{\alpha_d n})$ *for a given* $\alpha_d \geq 1/2$. *Then, it can be computed on graphs whose average degree is at least* $d-1$ *within time* $O^*(2^{\alpha_d n + \frac{(1-\alpha_d)(2m-(d-1)n)}{d+1}})$.

Proof Let $T(m,n)$ be the running time of the algorithm on graphs of order n with m edges. We first need to prove that T increases with m (this is trivial) and with n. Indeed:

$$\frac{\partial(\alpha_d n + \frac{(1-\alpha_d)(2m-(d-1)n)}{d+1}}{\partial n} = \alpha_d - \frac{(1-\alpha_d)(d-1)}{d+1}$$
$$= \frac{2d\alpha_d - d + 1}{d+1} \geq \frac{1}{d+1} > 0$$

We now proceed by induction on m, n. Trivially, the hypothesis of the statement holds for $m_0 = (d-1)n/2$. Suppose that it is true for any pair $n' < n, m' < m$. Since the graph has average degree greater than $d-1$, there exists some vertex of degree d or more. When branching on it, we get:

$$T(m,n) \leq 2T(m-d, n-1) \leq 2^{1+\alpha_d(n-1)+\frac{(1-\alpha_d)(2(m-d)-(d-1)(n-1))}{d+1}}$$
$$= 2^{\alpha_d n + \frac{(1-\alpha_d)(d+1)+(1-\alpha_d)(2(m-d)-(d-1)(n-1))}{d+1}}$$
$$= 2^{\alpha_d n + \frac{(1-\alpha_d)(2(m-d)-(d-1)(n-1)+(d+1))}{d+1}}$$
$$= 2^{\alpha_d n + \frac{(1-\alpha_d)(2m-(d-1)n)}{d+1}}$$

that completes the proof. □

As a straightforward consequence of the above lemma, the following proposition holds:

Proposition 2 *Assume that some problem that matches vertex branching can be computed on graphs with average degree at most* $d-1$, $d \in \mathbb{N}$, *in time* $O^*(2^{\alpha_d n})$ *for a given* $\alpha_d \geq 0.5$. *Then, it may be computed on graphs whose average degree is at most* d *within time* $O^*(2^{\alpha_{d+1} n})$, *where* $\alpha_{d+1} = \frac{d\alpha_d+1}{d+1}$.

Using Turán's Theorem which states that "for a graph of average degree d the size α^* of a maximum independent set verifies $\alpha^* \geq \frac{n}{d+1}$" the following theorem can be proved.

Theorem 3 *Any problem that is polynomial on totally disconnected graphs and matches vertex branching can be solved on graphs of average degree at most d with running time* $O^*(2^{dn/(d+1)})$.

Proof One can run some exact algorithm for MAX INDEPENDENT SET (for instance, the one in Fomin and Hoie (2006), that has complexity bounded by $O^*(2^{0.288n})$). Let S^* be the computed solution. Branch on any vertex belonging to its complementary, $V \setminus S^*$; the remaining graph is totally disconnected, so it can be solved in polynomial time. Thus, the total running time will be in $O^*(2^{0.288n} + 2^{n-\alpha(G)}) \subset O^*(2^{\frac{d}{d+1}n})$. □

Notice that the bound given by Turán's Theorem is tight for some problems that fit Definition 1. If the problem has the worst possible recurrence, then:

$$\text{opt}(G, w) = \max\{f_1(\text{opt}(G \setminus \{v\}, w_1)), f_2(\text{opt}(G \setminus \{v\}, w_2))\}$$

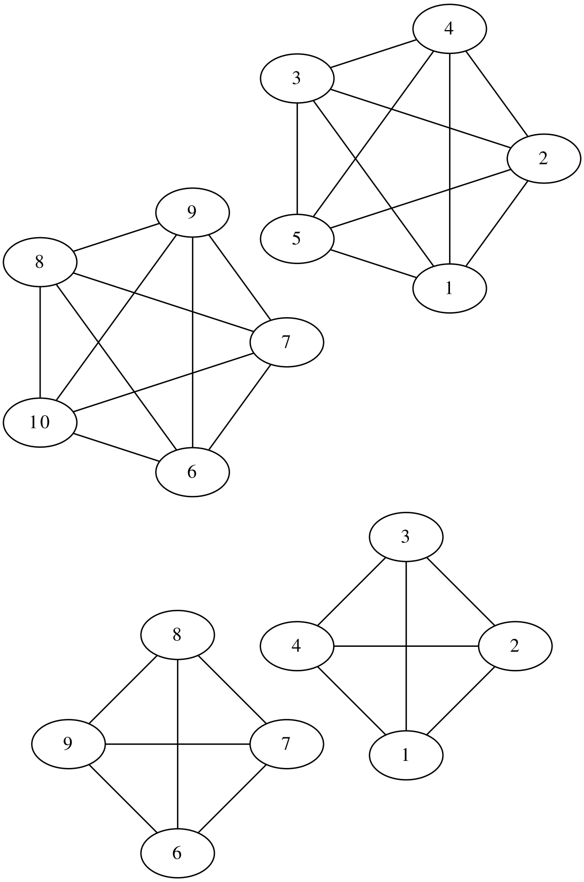
For instance, this is the case of maximum quasi-independent set. If we either make a greedy choice for the branching, i.e., if we always branch on a vertex of maximal degree, or we select an independent set of maximal size, then there exists an instance where the running time is at least $2^{\frac{d}{d+1}n}$. To see this, consider for any $\delta \leq d$ the graph G_δ that is composed of $n/(d+1)$ cliques of size $\delta+1$ (see Fig. 1). $T(G_1) = 2^{n/(d+1)}$. The algorithm removes one vertex in each connected component; we so have $T(G_\delta) = 2^{n/(d+1)} T(G_{\delta-1})$ and finally $T(G_d) = 2^{dn/(d+1)}$. On the other hand, $\alpha(G_d) = n/(d+1)$ (one vertex per clique).

Unfortunately, there is little hope for generalizing this corollary, since, unless **P** = **NP** no problem is polynomial on graphs of average degree bounded above by some $d > 0$, unless it belongs to **P** (just add some independent set to decrease d). Furthermore, restricting the instance set to graphs without isolated vertices, or even to connected graphs, does not help much, since the greedy branching may disconnect the graph as well. On the other hand, some improved results can be obtained for graphs of bounded *maximum* degree.

4.2.2 Using polynomial solution on graphs of maximum degree 2

Many problems that match vertex branching are in fact well-known to be polynomial on graphs of maximum degree 2, for instance MAX INDEPENDENT SET (or equivalently MAX CLIQUE and MIN VERTEX COVER). For some difficult problems like

Fig. 1 $K_{\delta+1}^{n/(d+1)}$ becomes $K_{\delta}^{n/(d+1)}$ after $n/d+1$ iterations



MAX QUASI-INDEPENDENT SET this remains true, but it is not straightforward. The corresponding result is stated in Subsect. 4.3 (Proposition 8).

Proposition 3 *Any problem that is polynomial on graphs of maximum degree* 2 *and matches vertex branching can be solved on graphs of average degree d with running time* $O^*(2^{dn/6})$. *This bound is tight for* $d = 3$ *and a greedy choice of the branching.*

Proof If there is no vertex of degree 3 or more, the problem can be solved in polynomial time. Otherwise, when branching on some vertex of maximum degree, we remove at least 3 edges from the graph, that means $T(m) \leq 2T(m-3)$, and leads to $T(m) = O^*(2^{m/3}) = O^*(2^{dn/6})$.

To show the tightness, consider a graph composed of $n/6$ copies of $K_{3,3}$. In any connected component, the algorithm greedily branches on three vertices, so $T(n) = 2^{n/2}$. □

Proposition 4 *Any problem that is polynomial on graphs of maximum degree* 2 *and matches vertex branching can be solved on graphs of average degree that rounds up*

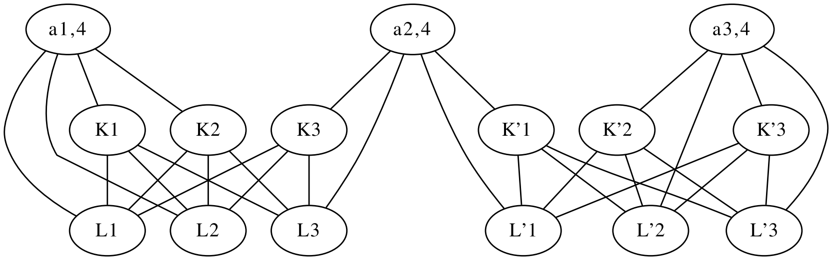


Fig. 2 Tightness example for $\Delta = 4$

to $d \leq 2$ with running time $O^*(2^{\frac{d-1}{d+1}n})$. If the average degree is d, this bound is tight for a greedy choice of the branching.

Proof In a notation like the one introduced in Proposition 2, this means that we claim $\alpha_{d+1} = (d-1)/(d+1)$. As a consequence of Proposition 3 with $d \in \{2,3\}$, we see that any problem which is polynomial on graphs of maximum degree 2 and matches vertex branching can be solved on graphs of average degree 2 (resp., 3) within time $O^*(2^{n/3})$ (resp., $O^*(2^{n/2})$). Thus, our hypothesis is verified for $d = 2$ and 3.

Now assume that the statement holds for $d-1$. Then, according to Proposition 2, $\alpha_{d+1} = \frac{d\frac{d-2}{d}+1}{d+1} = \frac{d-1}{d+1}$, and the result yields by recurrence.

In order to prove tightness, we form the graph G'_δ in the following way (Fig. 2):

- G'_3 is composed of $\frac{2n}{3(d+1)}$ copies of $K_{3,3}$. $T(G'_3) = 2^{2n(d+1)}$.
- Partition each pair of copies of $K_{3,3}$ into three subsets of size 4, namely A_1, A_2, A_3. For any i add a vertex $a_{i,4}$ adjacent to all the vertices in A_i. That is G'_4.
- For any $\delta \leq d-1$, form $G'_{\delta+1}$ by adding a vertex $a_{i,d+1}$ adjacent to all the vertices in $A_i \cup \{a_{i,4}, \ldots, a_{i,d}\}$.

The algorithm removes three vertices from each connected component, so we have $T(G'_\delta) = 2^{n/(d+1)} T(G'_{\delta-1})$ and finally $T(G'_d) = 2^{(d-1)n/(d+1)}$. □

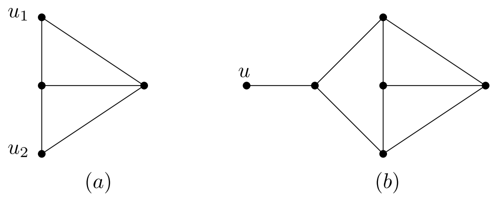
4.2.3 *Performance of bottom-up for problems that match vertex branching, on graphs of bounded maximum degree*

We now study the performance of bottom-up algorithms for problems that match vertex branching, on graphs of bounded maximum degree. We first deal with the case of graphs of maximum degree 3, followed by the general case of graphs with bounded maximum degree.

Proposition 5 *Any problem that matches vertex branching and is polynomial on graphs of maximum degree* 2 *can be solved on graphs of maximum degree* 3 *within time $O^*(2^{3n/8})$.*

Proof Notice that the bound claimed is lower than the $O^*(\sqrt{2}^n)$ on graphs of average degree 3 or less from Proposition 4. In our analysis, we use the following result,

Fig. 3



established by Reed (1996): *On graphs of minimum degree* 3*, the size of* MIN DOMINATING SET *is not greater than* $3n/8$.

We consider the vertices of the input graph of degree 2 or less which appear in connected components that contain at least a vertex of degree 3. We complete the graph by adding fictive edges between them until they all have degree at least 3. Notice that in the following cases this completion will be not successfully finished:

(i) A vertex v of degree 1 remains. In this case we add the gadget shown in Fig. 3(a) and the edges (v, u_1) and (v, u_2). Thus, the number of the vertices of the graph becomes $n' = n + 4$.
(ii) A vertex v of degree 2 remains. In this case we add the gadget shown in Fig. 3(b), where v coincides with u. Thus, the number of the vertices of the graph becomes $n' = n + 5$.
(iii) Two adjacent vertices, v_1 and v_2, both of degree 2 remain. In this case we add the gadget shown in Fig. 3(b) and the edges (v_1, u) and (v_2, u). Thus, the number of the vertices of the graph becomes $n' = n + 6$.
(iv) Two adjacent vertices, v_1 and v_2, of degree 1 and 2, respectively, remain. In this case we add the gadget shown in Fig. 3(b), where v_1 coincides with u. Moreover, we add the gadget shown in Fig. 3(b) and the edges (v_1, u) and (v_2, u) Thus, the number of the vertices of the graph becomes $n' = n + 11$.

Note that no vertex that was already at the beginning of the process of degree 3 shall receive a new neighbor this way. Furthermore, the number of vertices n' of the new graph is at most $n + 11$.

Then, we run an exact algorithm to find a MIN DOMINATING SET on the modified graph. This subset, namely $\mathcal{D}$, has a size at most $3n'/8 \leq 3n/8 + o(1)$. $\mathcal{D}$ is perhaps not a dominating set in G, but it is adjacent to all vertices of degree 3. In other words, once we have branched on any $v \in \mathcal{D}$, the remaining graph contains only vertices of degree 2 or less.

Of course there is an additive factor to our running time, that is the complexity of solving MIN DOMINATING SET on a graph where any vertex except a finite number of them have degree 3. Using the algorithm by Fomin and Hoie (2006), we see that such a set can be found in $O^*(2^{0.265n}) \subset O^*(2^{3n/8})$. □

A rather immediate corollary is that any problem that matches vertex branching and is polynomial on graphs of maximum degree 2 can be solved on graphs of maximum degree Δ with running time $O^*(2^{n(1-(5/8)^{\Delta-2})})$. Nevertheless, for $\Delta \geq 4$, the result from Proposition 4 overlaps this one.

In the case where we can only make the weaker hypothesis that the problem is polynomial on totally disconnected graphs (in fact, we need a somewhat stronger hypothesis, namely, polynomiality on collection of bounded cliques), it is still possible to improve the $O^*(2^{3n/4})$ result from Theorem 3, if we know that our graph has maximum degree 3 instead of average degree 3 or less:

Proposition 6 *Any problem that matches vertex branching and is polynomial on collections of cliques of bounded cardinality can be solved on graphs of maximum degree* 3 *within time* $O^*(2^{2n/3})$.

Proof If $G(V, E)$ contains a collection of 4-cliques, namely K, we first consider $G'(V', E') = G[V \setminus K]$. We can find in polynomial time a 3-coloring of G'. One of the three colors is an independent set S of size at least $|V'|/3$. Any possible subset of $V' \setminus S$ can be tested within time $O^*(2^{2|V'|/3}) \subset O^*(2^{2n/3})$, and the remaining graph is a collection of cliques of size 1 or 4. □

Proposition 7 *Any problem that matches vertex branching and is polynomial on graphs of maximum degree* 2 *can be solved on graphs of average degree* $d \leq 3$ *with running time* $O^*(2^{21n/46})$.

Proof If $\Delta \leq 3$, the proposition is a consequence of Proposition 5. Otherwise we perform a sequence of branchings, at each time choosing a vertex of maximal degree, until our graph have maximum degree 3. Then, we consider the size of the remaining graph:

(i) If $n' < 20n/23$, we greedily branch on vertices of degree 3 until the graph has maximum degree 2.
(ii) Otherwise, we compute a MIN DOMINATING SET as described on Proposition 5 and branch on any vertex of it.

We form the finite sequence $\{n_\Delta, n_{\Delta-1}, \ldots, n_4\}$, where n_i is the number of vertices of degree i we branch on during first step of our algorithm. Fix $\sigma = \sum_{i\geq 4} n_i$. Since i is the degree at the time we branch, not in the initial graph, the number of deleted edges is $\sum_{i\geq 4} i n_i \geq 4\sigma$.

If hypothesis (i) holds, our algorithm is within time $O^*(2^x)$, where $x \leq \sigma + \frac{m-4\sigma}{3} \leq \frac{n}{2} - \frac{1}{3} \times \frac{3n}{23} = \frac{21n}{46}$.

On the other hand, if hypothesis (ii) is true, the time of the algorithm is within $O^*(2^{x'})$, and $x' \leq \sigma + \frac{3(n-\sigma)}{8} \leq \frac{3n}{8} + \frac{5}{8} \times \frac{3n}{23} = \frac{21n}{46}$, that completes the proof. □

Thus, the following theorem holds:

Theorem 4 *Any problem that matches vertex branching and is polynomial on graphs of maximum degree* 2 *can be solved on graphs of average degree bounded above by* d *within time* $O^*(2^{\frac{d-27/23}{d+1}n})$, *for any* $d \geq 3$.

Proof The case $d = 3$ is nothing but Proposition 7. Assume that it is true for all values of average degree less than or equal to $d - 1$. Then, thanks to Proposition 2, we can compute a solution when the average degree is less than or equal to d, within time $O^*(2^{\alpha_{d+1} n})$, where $\alpha_{d+1} = \frac{d\alpha_d + 1}{d+1} = \frac{d\frac{d-1-27/23}{d}+1}{d+1} = \frac{d-27/23}{d+1}$. By induction, this is true for any d. □

4.3 Applying the bottom-up scheme for exact solution of the f-MAX QUASI-INDEPENDENT SET

Next proposition establishes the possibility to use a vertex branching method directly derived from the bottom-up scheme, for finding an optimal f-quasi-independent set within the time stated in Theorem 4.

Proposition 8 MAX QUASI-INDEPENDENT SET *is polynomial on graphs of maximum degree* 2 *or less.*

Proof Let Δ be the maximum degree of the graph we consider. If $\Delta = 0$, we add vertices with minimum weight in a greedy manner, until condition $w_0 + \sum_{v \in Q} w(v) \leq f(|Q| + q_0)$ gets violated, and find the optimal. If some vertex v has degree 1, let u be its only neighbor.

- If $\delta(u) = 1$, and, say, $w(u) \geq w(v)$, we can assume u belongs to the solution only if v does. Thus, we may remove edge $\{u, v\}$ and increase $w(u)$ by one. Of course this is symmetric between v and u.
- If $\delta(u) = 2$ and $w(u) \geq w(v)$, we can *a fortiori* remove the edge and increase $w(u)$ by 1.
- Otherwise, $\delta(u) = 2$ and $w(u) \leq w(v) - 1$. Since $w(u)$ might increase by at most 1 (if the second neighbor of u belongs to the sought optimal), we can safely remove $\{u, v\}$ and increase $w(v)$ by 1.

One can also consider the case where no vertex has degree 1, meaning that G is a set of cycles (at least one) and isolated vertices. Let $\{a_i\}_{i \leq k}$ be one of these cycles. If there exists some i where $w(a_i) \leq w(a_{i+1}) - 1$, then we can remove $\{a_i, a_{i+1}\}$ and add 1 to $w(a_{i+1})$. On the other hand, if for any i holds that $w(a_i) = w(a_{i+1})$, then all vertices in the cycle are identical, therefore we can decide arbitrarily to remove $\{a_1, a_2\}$ and increase $w(a_1)$ by 1. In any case, the cycle will become a path, and thus we can remove all its edges by successively disconnecting leaves, as explained above. □

From the discussion made in this Section, the following result is immediate.

Theorem 5 *Optimal* MAX QUASI-INDEPENDENT SET-*solutions in graphs of average degree* $\leq d$ *can be found in time* $O^*(2^{\frac{d-27/23}{d+1}n})$.

For instance, MAX QUASI-INDEPENDENT SET on graphs of average degree 3 can be solved in time $O^*(2^{\frac{21}{46}n})$, while in graphs of average degree 4, the corresponding time is $O^*(2^{\frac{13}{23}n})$.

5 Approximation algorithms

Let us first do a preliminary remark: MAX QUASI-INDEPENDENT SET problems being generalizations of MAX INDEPENDENT SET, the inapproximability results of the latter are immediately transferred to the former. In what follows, in this section, we study the approximation performance of several approximation algorithms for γ- and k-MAX QUASI-INDEPENDENT SET problems.

5.1 Approximation of γ-MAX QUASI-INDEPENDENT SET

5.1.1 When γ is bounded from below by a fixed constant

In this case, things are rather optimistic, since the following result holds.

Theorem 6 *Consider the γ-MAX QUASI-INDEPENDENT SET problem when γ is bounded below by a positive constant c. For any fixed $k \leq \Delta$, a solution of size at least k/Δ times the optimal can be computed in polynomial time.*

Proof Assume, w.l.o.g., that $\Delta > 2$ and the graph of the instance is not a collection of cliques (otherwise, the optimum can be found in polynomial time).

Under these assumptions, there is always a Δ-coloring of the graph and it can be found in polynomial time. Let S be a color class of the greatest size s; recall that $s \geq n/\Delta$. If $s \geq k/c$, then S will be a γ-quasi-independent set with the desired property, since by Lemma 2 we get $\frac{q^*}{s} \leq \frac{\Delta/\gamma}{k/c} \leq \frac{\Delta/c}{k/c} = \frac{\Delta}{k}$, where q^* is the size of the optimal Q^*.

Otherwise, we can enumerate all subsets of size ks or less within polynomial time, since S has bounded finite size and k is fixed by definition. If $q^* < ks$, we come with the optimal; else, by Lemma 1 we know that there exists a γ-quasi-independent set $Q \subset Q^*$ of size $q = ks$. For that set, we have $\frac{q^*}{q} \leq \frac{n}{ks} \leq \frac{n}{k(n/\Delta)} = \frac{\Delta}{k}$, and the result yields. □

5.1.2 A greedy algorithm

In this subsection, a greedy algorithm for computing a γ-quasi-independent set is discussed; the solution is initialized to some independent set S, and at each step a vertex of minimum degree to the current solution is being inserted; the insertions keep on, until the largest solution, respecting the sparsity specification, is reached.

procedure γ_QIS ($G(V,E)$: graph, $0 \leq \gamma \leq 1$: real, $S \subseteq V$: some independent set)
  $Q \leftarrow S$;
  $Q' \leftarrow S$;
  while ($|Q'| \leq |V|$)
    pick $v \in V \setminus Q'$ such that $\delta_{Q'}(v) = \min$ (break ties arbitrarily)
    $Q' \leftarrow Q' \cup \{v\}$;
    if ($|E[Q']| \leq \gamma\binom{|Q'|}{2}$)
      $Q \leftarrow Q'$;
    endif;
  endwhile;
  return(Q);
end γ_QIS

Obviously, γ_QIS always returns a solution, if S is set to some γ-quasi-independent set (any independent set in G, for instance the empty set, would do).

As it has already been mentioned, the non-hereditary character of γ-MAX QUASI-INDEPENDENT SET is reflected to the algorithm by the fact that some Q' produced during the execution of the algorithm may be infeasible while after some later vertex-insertions it may become feasible. This non-hereditary character of the problem is a major difficulty for a more refined analysis of algorithm γ_QIS. The following lemma gives a lower bound on the size of the solutions returned by this algorithm.

Lemma 5 *Let q be the size of the γ-quasi-independent set returned by the algorithm, where S has been initialized to some independent vertex set. It holds that $q > \frac{\alpha-1}{\sqrt{1-\gamma}}$ where α is the size of Q during the last step of the algorithm's execution before the first edges insertion.*

Proof Let $|E[Q]|$ be the number of edges in the solution Q. Denote by Q_i the state of Q at the moment v_i has been inserted in Q. By the definition of algorithm γ_QIS, if S is initialized to an independent set, the first α vertices inserted into the solution form a maximal independent set. Then $q = \alpha + \kappa$, for some κ, $0 \le \kappa \le n - \alpha$, i.e., after the insertion into the solution of the first α vertices which are independent of each other, $v_1, \dots, v_\kappa$ have been inserted. Let $\delta_{Q_i}(v_i)$ be the degree of v_i to Q_i. Then, $|E[Q]| = \sum_{i=1}^{\kappa} \delta_{Q_i}(v_i)$.

Notice that $\delta_{Q_1}(v_1) \le \alpha$ and $\forall i, 2 \le i \le \kappa$, $\delta_{Q_i}(v_i) \le \alpha + i - 1$; hence, it holds that $|E[Q]| \le \sum_{i=1}^{\kappa}[\alpha + (i-1)] = \kappa\alpha + \frac{\kappa(\kappa-1)}{2}$.

Assume, w.l.o.g., that the solution computed is not the whole graph (in which case it is trivially optimal). Then, by the definition of the algorithm γ_QIS, for any other candidate solution Q' containing Q and having size $\alpha + \kappa + l$ for some $l \ge 1$, we have:

$$\gamma \frac{(\alpha+\kappa+l)(\alpha+\kappa+l-1)}{2} < |E[Q']| = \sum_{i=1}^{\kappa+l} \delta_{Q_i}(v_i)$$

$$\le \sum_{i=1}^{\kappa+l}[\alpha + (i-1)] = (\kappa+l)\alpha + \frac{(\kappa+l)(\kappa+l-1)}{2}$$

$$\Rightarrow \quad \gamma \frac{(\alpha-1+\kappa+l)(\alpha+\kappa+l)}{2} < (\kappa+l)\alpha + \frac{(\kappa+l)(\kappa+l-1)}{2} \tag{4}$$

Setting $a = \alpha - 1, k = \kappa + l$, inequality (4) is written:

$$\gamma(a+k)(a+k+1) < 2k(a+1) + k(k-1)$$

$$\Leftrightarrow \quad (1-\gamma)k^2 + [2(1-\gamma)a + 1 - \gamma]k - \gamma a(a+1) > 0$$

$$\Leftrightarrow \quad (1-\gamma)k^2 + (1-\gamma)(2a+1)k - \gamma a(a+1) > 0 \tag{5}$$

Solving inequality (5) with respect to k, we get, after some easy algebra:

$$\begin{aligned} k &> -a - \frac{1}{2} + \frac{\sqrt{(1-\gamma)^2(2a+1)^2 + 4(1-\gamma)\gamma a(a+1)}}{2(1-\gamma)} \\ &= -a - \frac{1}{2} + \frac{\sqrt{4a^2+4a+1-\gamma}}{2\sqrt{1-\gamma}} \\ &> -a - \frac{1}{2} + \frac{\sqrt{4a^2+4a\sqrt{1-\gamma}+1-\gamma}}{2\sqrt{1-\gamma}} \\ &= -a - \frac{1}{2} + \frac{(2a+\sqrt{1-\gamma})}{2\sqrt{1-\gamma}} = -a + \frac{a}{\sqrt{1-\gamma}} \end{aligned} \tag{6}$$

Righthand-side inequality in (6) yields finally

$$\kappa + l + \alpha > \alpha - (\alpha - 1) + \frac{\alpha - 1}{\sqrt{1-\gamma}} \Leftrightarrow q = \kappa + \alpha > 1 - l + \frac{\alpha - 1}{\sqrt{1-\gamma}} \tag{7}$$

with the last inequality in (7) holding also for $l = 1$; hence, $q > \frac{\alpha-1}{\sqrt{1-\gamma}} \Rightarrow q \geq \lceil \frac{\alpha-1}{\sqrt{1-\gamma}} \rceil$, that completes the proof. □

Combining Lemma 5 and item (1) of Lemma 2, we finally get:

Theorem 7 *For the* γ-MAX QUASI-INDEPENDENT SET *problem it is possible to find in polynomial time a solution of size* q *achieving approximation ratio* $\frac{q^*}{q} \leq \frac{(\Delta+1)\alpha_{\min}}{\alpha-1}\sqrt{1-\gamma}$ *where* q^* *is the size of an optimal quasi-independent set and* $\alpha_{\min}$ *the size of a minimum independent dominating set of the input graph. This ratio tends to* $(\Delta+1)\sqrt{1-\gamma}$.

5.1.3 *Moderately exponential approximation for* γ-QUASI-INDEPENDENT SET

We complete the approximation section for γ-MAX QUASI-INDEPENDENT SET by showing how it can be approximated within any constant ratio by exponential algorithms with running time better than that of an exact computation.

Theorem 8 *For any* $k \geq 1$, *it is possible to compute a* γ*-quasi-independent set of size at least* $1/k$ *of the optimal, within time* $O^*(2^{(\log_2(k+1) - k/(k+1)\log_2 k)n})$.

Proof We enumerate every subset of size at most $n/(k+1)$ or at least $kn/(k+1)$, within time:

$$\begin{aligned} \sum_{i \leq n/(k+1)} \left[\binom{n}{i} + \binom{n}{n-i}\right] &\leq n\binom{n}{n/(k+1)} \\ &\leq n^2 \frac{n^n}{\left(\frac{kn}{k+1}\right)^{kn/(k+1)}\left(\frac{n}{k+1}\right)^{n/(k+1)}} \\ &\leq n^2\left(\frac{k+1}{k^{k/(k+1)}}\right)^n \end{aligned}$$

We return the minimal one which is a γ-quasi-independent set. If for the size of the optimal q^* it holds that $q^* \geq kn/(k+1)$ or $q^* \leq n/(k+1)$, we come with the optimal. Otherwise, by Lemma 1, we find a quasi-independent set Q of size $q \geq n/(k+1)$, that means $\frac{q^*}{q} \leq \frac{kn/(k+1)}{n/(k+1)} \leq k$, and the proof is completed. □

The following result exhibits a further link between γ-QUASI-INDEPENDENT SET and MAX INDEPENDENT SET.

Theorem 9 *Given some algorithm that computes an exact solution for* MAX INDEPENDENT SET *on G within time $O^*(c^n)$, for some constant c, a γ-quasi-independent set of size at least $1 + \gamma n/2$ can be computed within time $O^*(c^n)$.*

Proof Suppose, w.l.o.g., that $\gamma/2 < 1$, (otherwise the optimal is trivial). Consider an optimal γ-quasi-independent set Q^* of size q^*. We denote by $\alpha^*(G)$ the size of a maximum independent set in G. Applying Turán's Theorem (see, for example Berge 1973) to $G[Q^*]$, we get:

$$\alpha^*(G[Q^*]) \geq \frac{(q^*)^2}{|E[Q^*]| + q^*} \geq \frac{q^*}{\gamma(q^* - 1)/2 + 1} \geq \frac{2}{\gamma + (2 - \gamma)/q^*} \tag{8}$$

Hence, by (8), using the optimal solution returned by the algorithm as a γ-quasi-independent set, guarantees the ratio $\frac{q^*}{\alpha^*(G)} \leq \frac{q^*}{\alpha^*(G[Q^*])} \leq \frac{\gamma q^* + 2 - \gamma}{2} \leq \frac{\gamma n}{2} + 1$, that completes the proof. □

5.2 Approximation of k-MAX QUASI-INDEPENDENT SET

In this section we deal with polynomial approximation of k-QUASI-INDEPENDENT SET. We propose a greedy algorithm for that purpose, based upon the same idea as γ_QIS presented above; however, k-MAX QUASI-INDEPENDENT SET being a hereditary problem, the algorithm stops as soon as it finds the first vertex whose insertion violates the condition on the number of edges allowed in the solution.

Procedure k_QIS($G(V, E)$: Graph, k: integer≥ 0, S: some independent set)
  $Q \leftarrow S$;
  while ($|E(Q)| \leq k$)
    pick $v \in V \setminus Q$ such that $\delta_Q(v) = \min$ (break ties arbitrarily)
    $Q \leftarrow Q \cup \{v\}$;
  endwhile
  return($Q \setminus \{v\}$);
end k_QIS

In what follows, for a vertex set S, we define $\theta(S) = \max_{v \in V \setminus S}\{\delta_S(v)\}$.

Theorem 10 *For the k-*MAX QUASI-INDEPENDENT SET *problem it is possible to find in polynomial time a solution of size q achieving approximation ratio*:

$$\frac{q^*}{q} \leq \frac{\alpha^*\theta(S) + k\theta(Q)}{\alpha\theta(Q) + k} \leq \max\left\{\frac{\alpha^*}{\alpha}, \theta(Q)\right\}$$

where q^* is the size of the optimal, α^* is the size of a maximum independent set in G and α the size of some independent set.

Proof Let Q be the solution computed by the algorithm; as in the proof of Lemma 5, we note by q its size, and write $q = \alpha + x$, with α the size of Q at the last step before the first edge insertions performed by the algorithm; we can suppose, w.l.o.g., that this set is already computed before starting the algorithm. Clearly, for each of the remaining x vertices, at most $\theta(S)$ edges are inserted into the quasi-independent set; thus, with Q_i be the solution after the i-th from the x vertices insertions that brought some edges in, we have:

$$k = \sum_{i=1}^{x} \delta_{Q_i}(v_i) \leq \theta(S)x \Rightarrow k \leq \theta(S)(q-\alpha) \Leftrightarrow q \geq \frac{k}{\theta(S)} + \alpha \tag{9}$$

Let Q^* be an optimal solution with $|Q^*| = q^*$ and consider the graph $G[Q^*] = (Q^*, E')$. Obviously, $|E'| \leq k$. The set Q^* can be seen as the union of two sets S^* and T, where S^* is a maximum independent set of $G[Q^*]$ and $T = Q^* \setminus S^*$. On the other hand, E' is the union of the set E'_T of edges within the set T and of the set $E'_{S^*,T}$ of edges between S^* and T. Obviously, $|E'_{S^*,T}| \leq k$. Moreover, since the graph $(Q^*, E'_{S^*,T})$ is bipartite and connected (the set S^* is maximal for inclusion in both $G[Q^*]$ and $(Q^*, E'_{S^*,T})$), the fact that $|E'_{S^*,T}| \leq k$ implies $|T| \leq k$. So, $q^* \leq |S^*| + k \leq \alpha^* + k$, where α^* is the size of a maximum independent set in the graph G. Combining this bound with (9), we finally get the result. □

Recall that the best approximation ratio (as function of Δ) known for MAX INDEPENDENT SET is $(\Delta+2)/3$ and is guaranteed by the natural greedy MAX INDEPENDENT SET-algorithm (Halldórsson and Radhakrishnan 1994).

Suppose first that $\theta(Q) \geq 3$. Then, by item (2) of Lemma 2, the approximation ratio of algorithm k_QIS is bounded from above by $\Delta/3$. Assume now that $\theta(Q) \leq 3$. Then, by Theorem 10, the approximation ratio of the algorithm is bounded above by $\max\{\frac{\alpha^*}{\alpha}, 6\} \leq \frac{\alpha^*}{\alpha} \leq \frac{\Delta+2}{3}$, and the following holds.

Corollary 2 k-MAX QUASI-INDEPENDENT SET *is approximable in polynomial time within ratio* $(\Delta+2)/3$.

6 Some experimental results

6.1 Algorithm γ_QIS

Algorithm γ_QIS, presented in Sect. 5.1.2 has been run on 20 randomly generated graphs of each size (10, 20 and 30 vertices; edges in an instance have been generated with a probability p, $0.1 < p < 0.5$). Optimal solutions have been computed with the exact method of Sect. 4. Table 1 gives a summary of the experimental results obtained. It contains for every value of γ, the worst, best and average ratios and the percentage of optima returned by the algorithm.

One may remark that the more the density of the sought subgraph comes close to the density of the instance, the best the quality of the returned solution is.

Table 1 Experimental performance of algorithm γ_QIS

γ	Worst ratio	Best ratio	Average ratio	% optimal solutions
$0.2d(G)$	0.667	1	0.947	66.667%
$0.4d(G)$	0.7	1	0.969	75%
$0.6d(G)$	0.75	1	0.983	83.333%
$0.8d(G)$	0.905	1	0.996	93.333%
$1/n$	0.667	1	0.96	75%
$1/\sqrt{n}$	0.778	1	0.98	85%
$\log(n)/n$	0.778	1	0.973	76.667%

Table 2 Experimental performance of algorithm k_QIS

k	Worst ratio	Best ratio	Average ratio	% optimal solutions
$2\sqrt{m}$	0.93	1	0.99	90%
$\sqrt{n}$	0.83	1	0.98	80%
$\log(m)$	0.83	1	0.98	80%
$\log(n)$	0.83	1	0.97	75%
$m/2$	0.96	1	0.99	92.5%
$m/3$	0.96	1	0.99	95%
$n/2$	0.90	1	0.99	90%
$n/3$	0.90	1	0.99	87.5%

6.2 Algorithm k_QIS

Another set of tests have been implemented in order to experimentally observe the behavior of performance of algorithm k_QIS (devised in Sect. 5.2) for several values of k. As for the case of γ-MAX QUASI-INDEPENDENT SET, we have used the exact method of Sect. 4 to compute optimal solutions of the test instances. Instances have been generated randomly, following a probability p, $0.1 < p < 0.5$, for an edge to be present in the instance graph.

The tests indicate that the algorithm performs fairly well in small instances. A summary of the obtained results is given in Table 2.

7 Conclusions

In this paper, we have presented the quasi-independent set problem that consists of searching in a graph for a maximum size subgraph that satisfies a suitably defined sparsity condition; according to the definition of the latter, this problem is formalized as the f-MAX QUASI-INDEPENDENT SET problem, or the γ-MAX QUASI-INDEPENDENT SET, or finally k-MAX QUASI-INDEPENDENT SET problem. An exact method for computing an optimal f-MAX QUASI-INDEPENDENT SET-solution within time $O^*(2^{\frac{d-27/23}{d+1}n})$ in graphs of average degree bounded above by d has been designed, together with polynomial algorithms for finding approximate γ-MAX QUASI-INDEPENDENT SET- and k-MAX QUASI-INDEPENDENT SET-solutions, respectively. It has been shown that MAX QUASI-INDEPENDENT SET is polynomial in split graphs, while it is **NP**-hard in bipartite graphs. Determining its complexity in other graph-classes, mainly those where MAX INDEPENDENT SET is polynomial, is an interesting matter of ongoing research. The experimental results obtained suggest that the greedy algorithms for γ-MAX QUASI-INDEPENDENT SET and k-MAX QUASI-INDEPENDENT SET perform better than it can be shown by their theoretical analysis. However, we believe that in order to refine them, one has to develop non-combinatorial arguments; such a task is another promising perspective for future work.

Finally, other ideas of generalizing the approach of the greedy algorithms for γ-MAX QUASI-INDEPENDENT SET and k-MAX QUASI-INDEPENDENT SET could be tested. Indeed, these algorithms are based upon the fact that the (current) solution Q can be merged with another (in our case, a selected single vertex v) to form a new, larger one. As subtle point in algorithm γ_QIS is that it allows for "temporary" violations of the sparsity measure Q, if they can be repaired after a number of greedy vertex insertions. However, this is rarely the case: although it is possible to find such instances, none has been generated in experimental tests. A more general algorithm could be based on merging of a current solution Q with another, disjoint one Q'.

Acknowledgements The very pertinent and useful suggestions and comments of two anonymous referees are gratefully acknowledged.

References

Abello J, Resende MGC, Sudarsky S (2002) Massive quasi-clique detection. In: Proceedings of LATIN 2002. LNCS, vol 2286. Springer, Berlin, pp 598–612

Asahiro Y, Iwama K, Tamaki H, Tokuyama T (2000) Greedily finding a dense subgraph. J Algorithms 34(2):203–221

Berge C (1973) Graphs and hypergraphs. North-Holland, Amsterdam

Boginski V, Butenko S, Pardalos P (2005) Mining market data: a network approach. Comput Oper Res. Available online at http://www.sciencedirect.com

Corneil DG, Perl Y (1984) Clustering and domination in perfect graphs. Discrete Appl Math 9:27–39

Feige U, Kortsarz G, Peleg D (2001) The dense k-subgraph problem. Algorithmica 29(3):410–421

Fomin FV, Hoie K (2006) Pathwidth of cubic graphs and exact algorithms. Inf Process Lett 97(5):191–196

Goldberg AV (1984) Finding a maximum density subgraph. Technical report UCB CSD 84/171, University of California, Berkeley, CA

Halldórsson MM, Radhakrishnan J (1994) Greed is good: approximating independent sets in sparse and bounded-degree graphs. In: Proceedings of STOC 1994, pp 439–448

Hartwell LH, Hopfield JJ, Leibler S, Murray AW (1999) From molecular to modular cell biology. Nature 402:C47–C52

Hochbaum DS, Goldschmidt O (1997) k-edge subgraph problems. Discrete Appl Math 74(2):159–169

Jagota A, Narasimhan G, Šoltés Ľ (2001) A generalisation of maximal independent sets. Discrete Appl Math 109(3):223–235

Krishnamoorthy MS, Deo N (1979) Node-deletion NP-complete problems. SIAM J Comput 8:619–625

Picard JC, Queyranne M (1982) Selected applications of minimum cuts in networks. Inf Syst Oper Res 20:394–422

Reed B (1996) Paths, stars and the number three. Comb Probab Comput 5:277–295

Yannakakis M, Lewis J (1980) The node-deletion problem for hereditary properties is NP-complete. J Comput Syst Sci 20(2):219–230

Zuckerman D (2006) Linear degree extractors and the inapproximability of max clique and chromatic number. In: Proceedings of STOC 2006, pp 681–690