Independent sets with domination constraints

Magnús M. Halldórsson*
University of Iceland
Reykjavik, Iceland

Jan Kratochvíl†
Charles University
Prague, Czech Republic

Jan Arne Telle
University of Bergen
Bergen, Norway

Abstract

A ρ-independent set S in a graph is parameterized by a set ρ of natural numbers that constrains how the independent set S can dominate the remaining vertices ($\forall v \notin S : |N(v) \cap S| \in \rho$.) For all values of ρ, we classify as either $\mathcal{NP}$-complete or polynomial-time solvable the problems of deciding if a given graph has a ρ-independent set. We complement this with approximation algorithms and inapproximability results, for all the corresponding optimization problems.

The approximation results extend also to several related independence problems. In particular, we obtain a $\sqrt{m}$ approximation of the Set Packing problem, where m is the number of base elements, as well as a $\sqrt{n}$ approximation of the maximum independent set in the power graphs G^t, for t even.

1 Introduction

A large class of well-studied domination and independence properties in graphs can be characterized by two sets of nonnegative integers σ and ρ. A (σ, ρ)-set S in a graph has the property that the number of neighbors every vertex $u \in S$ (or $u \notin S$) has in S, is an element of σ (of ρ, respectively) [8]. This characterization facilitates the common algorithmic treatment of problems defined over sets with such properties. Unfortunately, the investigations of uniform complexity classification for this class of problems have so far all been incomplete [8, 6, 9]. In this paper we give a complete complexity classification, up to $\mathcal{P}$ vs. $\mathcal{NP}$, of the cases where $\sigma = \{0\}$, which constitute maybe the most important subclass of (σ, ρ)-problems.

This class of problems is precisely when the chosen set is to be independent. Independent (stable) sets in graphs are a fundamental topic with applications wherever we seek a set of mutually compatible elements. It is therefore natural to study the solvability of finding independent sets with particular properties, as in this case, where outside vertices are restricted in their domination.

Assume that we have an oracle for deciding membership in $\rho \subset \mathbb{N}$. Let $N(v)$ denote the set of neighbors of a vertex v. Consider the following decision problem:

ρ-IS Problem

*Adjunct affiliation: University of Bergen
†Research support in part by Czech research grants GAUK 194 and GAČR 0194/1996.

Given: A graph G
Question: Does G have an independent set of vertices $S \neq \emptyset$ with $|S| \geq \min\{k : k \notin \rho\}$ such that $\forall v \notin S : |N(v) \cap S| \in \rho$?

When ρ is the set of all positive integers the ρ-IS problem is asking for an independent dominating set, a problem which is easy since any maximal independent set is also a dominating set. When $\rho = \{1\}$ the ρ-IS problem is asking for the existence of a perfect code, a problem which is $\mathcal{NP}$-complete even for planar cubic graphs [5] and for chordal graphs [6]. The natural question becomes: For what values of ρ is the ρ-IS problem solvable in polynomial time? In the next section we resolve this question for all cases, up to $\mathcal{P}$ vs. $\mathcal{NP}$.

Approximation algorithms Even for the cases when the decision problem is solvable in polynomial time, the corresponding optimization problem, finding a minimum or maximum size ρ-IS, is hard. In section 3 we give on the one hand approximation algorithms for these optimization problems, and on the other hand strong inapproximability results.

The class of problems that we can approximate is that of finding an independent set where vertices outside the set are adjacent to at most a given number k vertices inside. We obtain performance ratios of $O(\sqrt{n})$ for these problems. This is significantly better than what is known for the ordinary Independent Set problem, where the best approximation is merely $O(n/\log^2 n)$ [1], a $\log^2 n$ factor from trivial. In fact, it is known that obtaining a performance ratio that is any fixed root of n factor better than trivial is highly unlikely [4]. We find that the same algorithm technique extends to a number of related independence problems, for which no non-trivial bounds were known.

Given a base set with m elements and a collection of n subsets of the base set, the Set Packing problem is to find the largest number of disjoint sets from the collection. There is a standard reduction from Independent Set to Set packing [2] where the number of sets n equals the number of vertices of the graph and the number of base elements m equals the number of edges of the graph. Thus, the hardness results of [4] translates to a $n^{1-\epsilon}$ lower bound for set packing, as a function of n, but only a $m^{1/2-\epsilon}$ lower bound in terms of m. This leaves a considerable gap in our understanding of the approximability of the problem in cases e.g. when m is linear in n.

We resolve this issue by giving a simple and efficient approximate algorithm that yields a performance ratio of $\sqrt{m}$. Our algorithm also yields an $O(\sqrt{m})$ performance ratio for the Maximum k-Matching of a set system (see definition in Section 3), and a $\sqrt{n}$ ratio for the maximum collection of vertices of a graph of mutually distance at least $t+1$, where t is even. In all of these cases, the bounds are essentially best possible.

2 Decision Problems

In this section we prove the following theorem.

Theorem 2.1 *The ρ-IS problem is $\mathcal{NP}$-complete if there is a positive integer $k \notin \rho$ with $k+1 \in \rho$, and otherwise it is solvable in polynomial time.*

The polynomial cases are summarized in the following result:

Lemma 2.2 *The ρ-IS problem is solvable in polynomial time if $\rho = \emptyset$, $\rho = \mathbb{N}^+$ or $\rho = \{0, 1, ..., k\}$ for some $k \in \mathbb{N}$.*

Proof. The cases $\rho = \emptyset$ and $\rho = \mathbb{N}^+$ are trivial. When $\rho = \{0, 1, ..., k\}$ for some $k \in \mathbb{N}$, we are asking if the input graph G has an independent set S of at least $k+1$ vertices such that every vertex not in S has at most k neighbors in S. The algorithm simply tries all subsets S of size $k+1$, and if none of them satisfy the conditions the answer is negative. ∎

We remark that when restricted to chordal graphs the ρ-IS problem is solvable in polynomial time whenever $\min\{k : k \in \rho\} \geq 2$ [6]. We turn to the $\mathcal{NP}$-complete cases, and first state two earlier results:

Theorem 2.3 *[8] The ρ-IS problem is $\mathcal{NP}$-complete whenever ρ is a finite nonempty subset of positive integers or when $\rho = \{k, k+1, ...\}$ for some $k \geq 2$.*

Theorem 2.4 *[5] Deciding if a cubic graph has a perfect code is $\mathcal{NP}$-complete.*

We first take care of an easy special case. The proof is omitted in this abstract.

Lemma 2.5 *Let k be a fixed integer. The ρ-IS problem with $\rho = \{0, k+1, k+2, k+3, \ldots\}$ is $\mathcal{NP}$-complete.*

Let EVEN be the set of all even natural numbers and ODD be the set of all odd natural numbers. As is often the case with parity problems, e.g. chromatic index of cubic graphs, the cases of EVEN-IS and ODD-IS require a special reduction for their $\mathcal{NP}$-completeness. The following result encompasses all $\mathcal{NP}$-complete cases except these two.

Lemma 2.6 *The ρ-IS problem is $\mathcal{NP}$-complete if ρ is not EVEN or ODD and there is a positive integer $k \notin \rho$ with $k+1 \in \rho$.*

Proof. Let $t = \min\{x : x \geq 1 \wedge x \in \rho \wedge x+1 \notin \rho\}$. If such t does not exist then either $\rho = \{k+1, k+2, ...\}$ and ρ-IS problem is $\mathcal{NP}$-complete by Theorem 2.3, or $\rho = \{0, k+1, k+2, ...\}$ and is $\mathcal{NP}$-complete by Lemma 2.5. Let $z = \min\{x : x > t \wedge x \notin \rho \wedge x+1 \in \rho\}$. If such z does not exist then $\rho = \{1, 2, ..., i\}$ and is $\mathcal{NP}$-complete by Theorem 2.3.

For any cubic graph G we construct a graph G' which has a ρ-IS if and only if G has a perfect code. We shall be assuming that G is sufficiently large, e.g. contain at least z^2 vertices.

A *claw* is a set of four vertices, consisting of a *center* vertex and three neighbors. The essential property we use is that exactly one vertex of a claw is contained in every perfect code of a cubic graph. Henceforth, when we refer to claws, we always mean the claws contained in copies of G in G'.

The derived graph G' consists of $z+1$ copies $G_1, ..., G_{z+1}$ of G, with vertices named $u_1, ..., u_{z+1}$ for $u \in V(G)$, along with a large collection of nodes connected into a clique. For each edge $uv \in E(G)$ add edges $u_i v_j$ for $1 \leq i, j \leq m$, to ensure that for any independent set S' in G' its projection S ($u \in S$ iff $\exists i : u_i \in S'$) onto G is also an independent set.

We now describe the 'mating' gadget that will ensure that a ρ-IS in G' contains either all copies of $u \in V(G)$ or no copies of u. For each vertex $u \in V(G)$ add $\lceil z/(z-t) \rceil$ new vertices to the clique, each vertex adjacent to u_1. We describe the additional neighbors for only one such vertex u_m: it is made adjacent to $u_2, ..., u_{z+1-t}$ and to t vertex-disjoint claws in G_1. The remaining new vertices will play the same role for the other copies of u.

There are two clique nodes for every group of $z+1$ vertex-disjoint claws in G'. These are connected to all the vertices of those claws in G', and to no other vertex in the copies of G. There are also two clique nodes for every group of t disjoint claws in G_1 or G_2, adjacent to the vertices of those claws in G_1 and G_2. To ease the presentation, we first prove two properties that must hold for any ρ-IS S of G', and then complete the specification of G' by adding some more vertices to the clique only.

Claim 1 *S contains a vertex in every claw in G', but contains no clique node.*

Proof. Recall that by definition a ρ-IS S must contain at least $t+1$ nodes. One of those nodes may be a clique node. Let q be the lesser of $|S|$ and $z+1$. Consider some q nodes Z from S. These can be covered by $z+1$ disjoint claws, possibly along with a single clique node. There exists a clique node v not in S that is adjacent to all these q nodes. To satisfy its degree constraint, it follows that q and $|S|$ are at least $z+1$.

Let Z be a set of $z+1$ nodes in S. We first argue that there are no more than two claws X and Y that are disjoint from each other and of distance three or more from claw-nodes in Z. Otherwise, first consider the case when Z contains a clique node w. Then, we can pick $z-1$ disjoint claws (some possibly containing no nodes in S) that cover exactly $z-1$ claw nodes from Z. The clique node that would be adjacent to those $z-1$ claws and to X and Y (and by definition to w) would be adjacent to exactly z nodes in S, failing the ρ-IS property. On the other hand, if Z contains no clique node, then we pick z disjoint claws containing exactly z nodes from Z, and the clique node adjacent to these and to X would also be adjacent to z nodes in S.

We now argue that no clique node can be in S. Otherwise, we could select t claws whose centers are in S. The corresponding clique node would then be adjacent to these centers and the clique node in S, for a total of $t+1$ adjacencies in S, failing the ρ-IS property.

Finally, since S contains no clique node, we can argue similarly that there does not exist a claw that is without a node in S. ■

Claim 2 *Each claw in G_1 and G_2 contains either one vertex in S or three vertices in S.*

Proof. Let X be any claw. Since G is sufficiently large, we can choose $t-1$ other claws such that there is no vertex that is adjacent to a vertex in two or more chosen claws. Further, these claws are not adjacent to X. For each of these $t-1$ claws, choose a vertex in S (which exists by the previous claim), and form a new claw with this vertex as its center. The new set of $t-1$ claws is disjoint, and contains exactly t vertices in S, by the independence property. Further, it is disjoint from X.

Consider the clique vertex that is adjacent to X and to the new set of $t-1$ claws. It is adjacent to $t-1$ nodes in S plus the one to three vertices in X that are in S. Since ρ does not contain $t+1$, it follows that X contains exactly one node or three nodes in S. ■

We complete the construction of G' in three different manners depending on which of the following three cases holds:

(i) $\exists w \geq 3 : w \notin \rho \wedge w - 2 \in \rho$

(ii) not (i) but $0 \in \rho \wedge 1 \in \rho \wedge 2 \notin \rho$

(iii) not (i) or (ii) but $\exists w \geq 2 : w \in \rho \wedge w - 2 \notin \rho$

If none of these cases hold, but z and t as defined earlier exist, then ρ must be EVEN or ODD. In case (i) we add a node to the clique for each group of $w - 2$ vertex disjoint claws in G_1, and make the node adjacent to these claws. In case (ii) we add a node to the clique for each pair of vertices in G_1 which are at distance 2 in G, and make the node adjacent to the pair. In case (iii) we add several nodes to the clique for each $x \in G$ whose neighbors a, b, c form an independent set and make these new clique nodes adjacent to x_1, to x_2, and to all vertices in G_1 at distance 2 from x. Let x have neighbors a, b, c and these in turn have additional neighbors $a1, a2, b1, b2, c1, c2$, respectively. We add extra neighbors in G_2 to the new clique node depending on the common identities of these six vertices:

- all different: add nothing,
- one triple and 3 one-tuples: add nothing,
- one pair and 4 one-tuples: add extra copy in G_2 of the pair,
- two pairs and 4 one-tuples: add extra copies in G_2 of both pairs,
- three pairs: add extra copies in G_2 of all three pairs, and
- one triple, one pair and one one-tuple: add extra copies in G_2 of the pair and the one-tuple.

Such a node for $x \in G$, with neighbors as above, is added for each group of $w - 2$ vertex disjoint claws in G_1, with these claws as additional neighbors.

Claim 3 *Each claw in G_1 contains exactly one vertex in S'.*

Proof. Let X be any claw in G_1. We show that in either of cases (i),(ii) or (iii) X does not contain three vertices in S'.

In case (i) we can find a group of $w - 3$ claws in G_1 whose centers are all in S'. The node adjacent to X and these claws would have w neighbors in S' if X had 3 neighbors in S'.

In case (ii) no two vertices in G_1 at distance-2 in G can both be in S' and hence no claw in G_1 can contain three vertices in S'.

In case (iii) some set of $w - 2$ claws have the central vertex chosen. Let X have center node x_1. The clique node added for x_1 and these $w - 2$ claws has at least $w - 2$ neighbors in S'. It is easy to check, in each of the separate cases of common identities above, if the claw X has three vertices in S' then none of the remaining neighbors of this clique node is in S', and if X has one vertex in S' then exactly two of the remaining neighbors of this clique node is in S'. Since $w \notin \rho$ X cannot have three vertices in S'. ∎

Now that we know that each claw has exactly one vertex in any ρ-IS S' it can be readily checked that the mating gadget ensures that S' contains either all copies of a vertex $u \in V(G)$ or none of these copies.

A perfect code S in G gives the ρ-IS $S' = \{u_1, ..., u_{z+1} : u \in S\}$, and vice-versa the projection of any ρ-IS in G' onto G is a perfect code. ∎

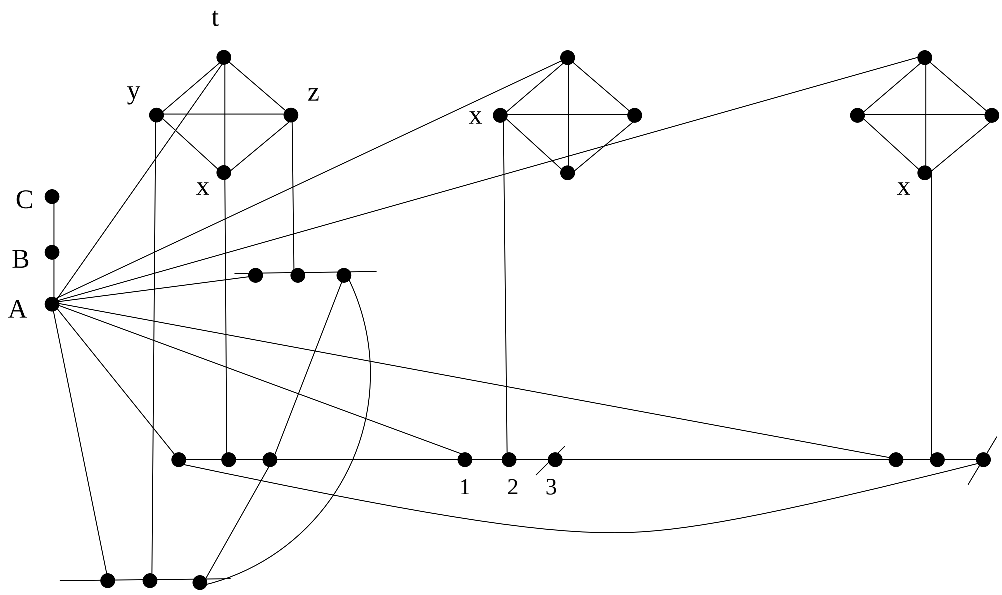


Figure 1: Part of the constructed graph G' for the EVEN-IS reduction. A triple t with elements x, y, z gives the upper-left 4-clique. Shown is the 9-cycle created by element x, and the 3-cycle created by x, y, z for this triple.

We continue by showing that the EVEN-IS problem is $\mathcal{NP}$-complete, by reduction from a $\mathcal{NP}$-complete case of problem EXACT COVER BY 3-SETS (X3C) [2].

Definition 2.7 *X3C-3: Given a system of triples s.t. every element of the base set belongs to exactly 3 triples, decide if there is a subset of triples s.t. every element belongs to exactly one of the chosen triples.*

Lemma 2.8 *The EVEN-IS problem is $\mathcal{NP}$-complete.*

Proof. We reduce from X3C-3, but consider instead the dual problem, which clearly has an exact cover iff the original one does: given a system of triples $T_1, ..., T_m$ drawn from a base set X, with every element of X belonging to exactly 3 triples, decide if there is a subset $I \subseteq X$ such that for each i, $|I \cap T_i| = 1$.

We construct a graph G such that G has an EVEN-IS iff the given system of triples has an exact cover. For each triple, say $T_i = \{x, y, z\}$, G contains a 4-clique with vertices t_i, t_i^x, t_i^y, t_i^z. For an element x appearing in the triples T_{x1}, T_{x2}, T_{x3}, G contains a cycle of 9 vertices: $x^1_{T_{x1}}, x^2_{T_{x1}}, x^3_{T_{x1}}, x^1_{T_{x2}}, x^2_{T_{x2}}, x^3_{T_{x2}}, x^1_{T_{x3}}, x^2_{T_{x3}}, x^3_{T_{x3}}$. For each element x and triple T_i that it appears in, G contains an edge between t_i^x and $x^2_{T_i}$. For a triple $T_i = \{x, y, z\}$, G contains a triangle on the three vertices $x^3_{T_i}, y^3_{T_i}, z^3_{T_i}$. Finally, G contains 3 additional vertices A, B, C with edges AB, BC and vertex A adjacent to vertex t_i for each triple and to each vertex $x^1_{T_i}$ for each element and triple that it appears in. See Figure 2.

For one direction of the proof, assume that I is a subset of elements such that for each i, $|I \cap T_i| = 1$. Then G has an EVEN-IS consisting of A, C, the vertices $t_i^x, x^3_{T_i}$ for the variable $x \in I \cap T_i$, and the vertices $x^2_{T_i}$ for the variables $x \in T_i$ that are not in I, for each i. It is easy to check that this is an EVEN-IS.

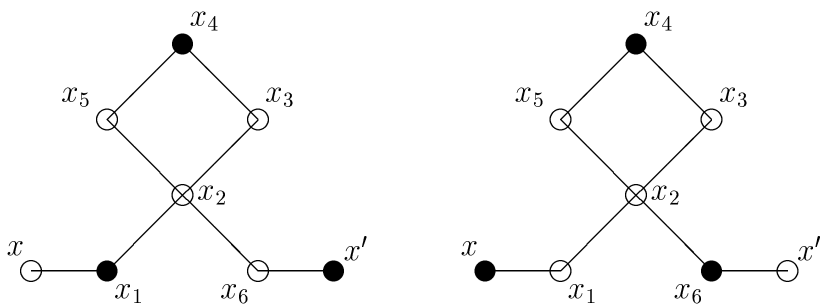
For the other direction of the proof, we first show that any non-empty EVEN-IS S in G must contain the vertex A. Otherwise, none of the vertices t_i^x can be in S, since t_i would have just one S-neighbor. Likewise, none of the vertices t_i can be in S, since then $x_{T_i}^2$ for each $x \in T_i$ would have to be in S to satisfy t_i^x. This in turn forces each alternating vertex around the cycle of vertices associated with the variable x to be in S. However, there are 9 such vertices, and we can only chose every alternating vertex of a cycle if the length of the cycle is even. Thus, if $A \notin S$ neither vertex of the 4-clique associated with a triple could be in S and thus neither could $x_{T_i}^2$ since t_i^x would then have only one S-neighbor. But neither can $x_{T_i}^1$ or $x_{T_i}^3$ since these would force some $x_{T_j}^2$ to be in S.

We thus know that any EVEN-IS S in G must contain the vertex A, and therefore in every 4-clique associated with a triple $T_i = \{x, y, z\}$ exactly one of t_i^x, t_i^y, t_i^z must be in S. Moreover, the corresponding element is said to be chosen, since if $t_{xi}^x \in S$ then $x_{T_{xi-1}}^3 \in S$ and also $t_{xi-1}^x \in S$, so that if an element is chosen in one 4-clique it is chosen in every 4-clique in which it appears. Since there is a 4-clique for each triple, we conclude that an EVEN-IS in G gives rise to an exact cover in the original triple system. ∎

Finally, we complete the proof of Theorem 2.1 by showing the $\mathcal{NP}$-completeness of ODD-IS.

Lemma 2.9 *The ODD-IS problem is $\mathcal{NP}$-complete.*

Proof. We reduce from the EVEN-IS problem. Note in the above proof that the EVEN-IS problem is $\mathcal{NP}$-complete for a graph on $13n + 3$ vertices, with the property that if it contains an EVEN-IS then it has size $3n + 2$. W.l.o.g. we let n be odd so that the input graph G to EVEN-IS has an even number of vertices and any EVEN-IS must have an odd number of vertices. We first describe a special gadget H_x.



For a vertex x, let H_x be a graph with vertices $x, x_1, \ldots, x_6, x'$ and edges xx_1, x_1x_2, x_2x_3, x_3x_4, x_4x_5, x_5x_2, x_2x_6, x_6x'. If $S \subset V(H_x)$ is an independent set such that every vertex $y \in \{x_1, \ldots, x_6\} - S$ has odd number of S-neighbors, then either $x, x_6 \in S$ (and $x_1, x' \notin S$), or $x_1, x' \in S$ (and $x, x_6 \notin S$). (Every such S must contain x_4 and exclude x_2, the rest then follows straightforwardly.)

Given a graph G subject to the EVEN-IS question, we construct G' with vertex set

$$V(G') = \bigcup_{x \in V(G)} V(H_x) \cup \{A\}$$

and edge set

$$E(G') = E(G) \cup \bigcup_{x \in V(G)} E(H_x) \cup \bigcup_{x \in V(G)} \{x'A\}.$$

We claim that G' has an ODD-IS if and only if G has a nonempty EVEN-IS.

Let $S \subset V(G)$ be a nonempty EVEN-IS in G, hence both $|S|$ and $|V(G)| - |S|$ are odd. Set $S' = S \cup \{x_6 : x \in S\} \cup \{x_4 : x \in V(G)\} \cup \{x', x_1 : x \in V(G) - S\}$. Every vertex $x \in V(G) - S$ has even number of S-neighbors plus the S'-neighbor x_1, thus an odd number of S'-neighbors. Vertices from $V(G') - (V(G) \cup \{A\})$ have obviously odd number of S'-neighbors and the S'-neighbors of A are $x', x \in V(G) - S$, whose number is odd as well.

Suppose on the other hand that S' is an ODD-IS in G'. We first argue that $A \notin S'$, since otherwise $x' \notin S'$ for every $x \in V(G)$, implying that $V(G) \subset S'$, a contradiction (we may assume wlog that G has at least one edge). Thus $A \notin S'$ and $|S' \cap \{x' : x \in V(G)\}|$ must be odd. Setting $S = S' \cap V(G) = \{x : x' \notin S'\}$, we deduce that $|S|$ is odd and hence S is nonempty. Straightforwardly, S is an EVEN-IS in G. ∎

3 Optimization

Let us consider the complexity of ρ-IS optimization problems. Clearly optimization is no easier than the corresponding decision problem, thus we are interested in the problems where the decision version is polynomial solvable. When an optimization problem turns out to be hard to compute, we would further like to know how hard it is to compute approximate solutions by polynomial-time algorithms.

We say that an algorithm *approximates* a problem *within* t if the solution computed on any instance never strays from the optimal by more than a multiplicative factor t. The algorithm then has *performance ratio* t. Note that the factor t may be a function of the size of the input. When a better approximation algorithm cannot be found, we naturally try to show that no better algorithm can be found given some natural complexity-theoretic assumption.

Minimization problems are trivial when ρ contains zero, which leaves the case $\rho = \mathbb{N}^+$ as the only non-trivial issue for minimization. This is the Minimum Independent Dominating Set problem, which is known to be $\mathcal{NP}$-hard to approximate within $n^{1-\epsilon}$, for any $\epsilon > 0$ [3]. The reduction holds even if the graph is sparse, thus it is hard within $m^{1-\epsilon}$. In fact, no sub-linear performance ratio is known for this problem.

The maximization problem with $\rho = \mathbb{N}^+$ is the Maximum Independent Set problem. The best performance ratio known is $O(n/\log^2 n)$ [1]. Håstad [4] has recently improved a sequence of deep results to show that this problem is hard to approximate within $n^{1-\epsilon}$, for any $\epsilon > 0$. This result is modulo the assumption that $\mathcal{NP} \neq \mathcal{ZPP}$, namely that zero-error randomized polynomial algorithms do not exist for all problems in $\mathcal{NP}$. This is highly expected, while slightly weaker hardness results are known under the stronger assumption that $\mathcal{P} \neq \mathcal{NP}$. We shall use this result in this paper, with the knowledge that weaker assumptions will then also transfer to our results. In particular, our reductions do give the $\mathcal{NP}$-hardness of the exact optimization problems considered.

The only remaining non-trivial maximization problems are the cases of $\rho = \{0, 1, \ldots, k\}$, for some $k \in \mathbb{N}$. We focus on these problems for the remainder of this section. We show them to be $\mathcal{NP}$-hard, and obtain nearly tight bounds on their approximabilities. The results are summarized in the following theorem. Let *opt* denote the size of the optimal solution of the instance.

Theorem 3.1 *The ρ-IS maximization problem with $\rho = \{0, 1, \ldots, k\}$, for some fixed $k \in \mathbb{N}$, can be approximated within $O(\sqrt{n})$ in polynomial time, but not within $n^{1/(k+1)-\epsilon}$ nor $opt^{1-\epsilon}$, for any fixed $\epsilon > 0$, unless $\mathcal{NP} = \mathcal{ZPP}$.*

Approximation algorithm We now give an algorithm that approximates some important problems on set systems. These results are interesting in their own right. Simple reductions then imply the same approximation for the ρ-IS problems with $\rho = \{0, 1, \ldots, k\}$.

Definition 3.2 *The* Set Packing *problem is the following: Given a base set S and a collection $\mathcal{C}$ of subsets of S, find a collection $\mathcal{C}' \subseteq \mathcal{C}$ of disjoint sets that is of maximum cardinality.*

Theorem 3.3 *Set Packing can be approximated within $\sqrt{m}$, where m is the size of the base set, in time linear in the input size.*

Proof. A greedy algorithm is given in Fig. 2. In each step, it chooses a smallest set and removes from the collection all sets containing elements from the selected set.

Greedy$(S,\mathcal{C})$
$t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  $X_t \leftarrow C \in \mathcal{C}$ of minimum cardinality
  $Z_t \leftarrow \{C \in \mathcal{C} : X \cap C \neq \emptyset \}$
  $\mathcal{C} \leftarrow \mathcal{C} - Z_t$
until $|\mathcal{C}| = 0$
Output $\{X_1, X_2, \ldots, X_t\}$

Figure 2: Greedy set packing algorithm

Let $M = \lfloor\sqrt{m}\rfloor$. Observe that $\{Z_1, \ldots, Z_t\}$ forms a partitioning of C. Let i be the index of some iteration of the algorithm, i.e. i is an integer, $1 \leq i \leq t$. All sets in Z_i contain at least one element of X_i, hence the maximum number of disjoint sets in Z_i is at most the cardinality of X_i. On the other hand, every set in Z_i is of size at least X_i, so the maximum number of disjoint sets in Z_i is also at most $\lfloor m/|X_i|\rfloor$. Thus, the optimal solution contains at most $\min(|X_i|, \lfloor m/|X_i|\rfloor) \leq \max_{x\in\mathbb{N}} \min(x, \lfloor m/x\rfloor) = M$ sets from Z_i.

Thus, in total, the optimal solution contains at most tM sets, when the algorithm finds t sets, for a ratio of at most M. ∎

The Strong Stable Set problem is the ρ-IS problem with $\rho = \{0, 1\}$. A strong stable set corresponds to a set of vertices in a graph of pairwise distance at least three. The Strong Stable Set problem reduces to Set Packing in the following way. Given a graph $G = (V, E)$, construct a set system $(S, \mathcal{C})$ with $S = V$ and $\mathcal{C} = \{N(v)|v \in V\}$. Then, a strong stable set corresponds to a set of nodes whose neighborhoods do not overlap, thus forming a set packing of $(S, \mathcal{C})$.

Corollary 3.4 *Strong Stable Set can be approximated within* $\sqrt{n}$.

The *Distance-t-IS* problem is that of finding the maximum number of vertices of mutual distance at least $t+1$ apart. It corresponds to finding an independent set in the power graph G^t (when given the root graph G). If A is the adjacency matrix of G and I is the identity matrix, then the adjacency matrix of G^t is obtained by computing $(A+I)^t$, replacing non-zero entries by ones, and eliminating self-loops. The Strong Stable Set problem is the problem of finding an independent set in G^2 (given G). Since the Distance-$2q+1$-IS is the problem of finding an independent set in $(G^q)^2$, it is a restricted case of the Strong Stable Set problem.

Corollary 3.5 *Let t be an odd natural number. The Distance-$2q+1$-IS problem can be approximated within* $\sqrt{n}$.

We now extend the application of the greedy set packing algorithm.

Definition 3.6 *A k-*matching *of a set system $(S, \mathcal{C})$ is a collection $\mathcal{C}' \subseteq \mathcal{C}$ such that each element in S is contained in at most k sets in $\mathcal{C}'$.*

In particular, a 1-matching is precisely a set packing.

Theorem 3.7 *Let k be an integer. The greedy set packing algorithm approximates the Maximum k-Matching problem within $k\sqrt{m}$.*

Proof. The sum of the sizes of sets in a k-matching is at most km. Thus, if each set contains at least q elements, then the matching contains at most $\lfloor \frac{km}{q} \rfloor$ sets.

Thus, for $i = 1, \ldots, t$, the optimal k-matching of Z_i contains at most $\lfloor \frac{km}{|X_i|} \rfloor$ sets. On the other hand, it never contains more than $k|X_i|$ sets from Z_i, since it contains at most k sets containing a particular element from X_i. Thus, it contains at most $k \min(|X_i|, m/|X_i|) = k\sqrt{m}$ sets from Z_i.

In total, the optimal k-matching contains at most $tk\sqrt{m}$ sets when the algorithm obtains t sets, for a performance ratio of $k\sqrt{m}$. ∎

Corollary 3.8 *Let k be a fixed integer. The ρ-IS problem, for $\rho = \{0, 1, \ldots, k\}$ is approximable within $O(\sqrt{n})$.*

Proof. Given an instance G to ρ-IS, form the set system of closed neighborhoods, as in the reduction of Strong Stable Set to Set Packing. Recall that the number of base elements m now equals the number of sets n. Clearly the solution output by the greedy set packing solution is a feasible solution, since it forms a $\{0,1\}$-IS.

Observe that any solution to the ρ-IS problem of G corresponds to a k-matching in the derived set system (while the converse is not true). Hence, by Theorem 3.7 the size of the algorithm's solution is also within $O(\sqrt{n})$ of the optimal ρ-IS solution. ∎

We note that the sizes of a maximum strong stable set and a maximum $\{0,1,2\}$-IS can differ by a factor of as much as $\Omega(\sqrt{n})$. Consider the graph G with x independent vertices $v_1, v_2, \ldots, v_x$, $\binom{x}{2}$ mutually adjacent vertices $w_{i,j}$, $1 \le i < j \le x$, with $w_{i,j}$ adjacent to v_s iff $s = i$ or $s = j$. Any pair of vertices in the graph have a common neighbor, thus no strong stable set can contain more than a single vertex. On the other hand, the v_i vertices form a $\{0,1,2\}$-IS. Thus, the ratio between these sizes is x, or about square root of the number of vertices.

Approximation lower bound A $(k+1)$-*uniform hypergraph* is a pair (V, E) where V is a set of vertices and E is a set of hyperedges (or subsets of V) of size $(k+1)$ each. A subset S of V is an independent set if no hyperedge is fully contained in S.

Our lower bound rests on the following reduction from the problem of finding an approximately maximum IS in a hypergraph.

Lemma 3.9 *If the ρ-IS maximization problem with $\rho = \{0, 1, \ldots, k\}$ can be approximated within $f(n)$, then the MIS problem in $(k+1)$-uniform hypergraphs can be approximated within $O(f(n)^{k+1})$.*

Also, if the former problem can be approximated within $g(opt)$, as a function of the optimal solution value opt, so can the latter.

Proof. Given a hypergraph H, construct a graph G as follows. G contains a vertex for each node and each hyperedge of H. The hyperedge-vertices form a clique, while the node-vertices are independent. A hyperedge-vertex is adjacent precisely to those node-vertices that correspond to nodes incident on the hyperedge.

We first claim that any independent set S in the hypergraph H is a ρ-IS in G. Clearly it is an independent set in G since it consists only of node-vertices. Each node-vertex thus has a ρ-value of 0. Hyperedge-vertices have exactly k node-vertices as neighbors and not all of those can be in S given the independence property of S in H. Thus, hyperedge-vertices have a ρ-value of at most $k-1$.

Any ρ-IS S in G can contain at most one hyperedge-vertex, and if we eliminate that possible vertex from S, it can be verified that the remainder corresponds to an independent set in H.

Taken together, any approximate solution to ρ-IS gives an equally approximate solution to MIS of H, within an additive one. Hence, ratios in terms of *opt* carry over immediately. For approximations in terms of the input size, we must factor in that $|V(G)| = |V(H)| + |E(H)| = O(|V(H)|^{k+1})$. ∎

To obtain the theorem, we need to show that MIS in hypergraphs is hard to approximate. We sketch here how the $n^{1-\epsilon}$ inapproximability result of [4] translates to the same bound for the case of uniform hypergraphs. Given a graph G, form a hypergraph H on the same vertex set, with hyperedges for any $(k+1)$-tuples such that some pair of vertices in the tuple form an edge in G. Then, we have a one-one correspondence between independent sets (of cardinality at least k) in G and in H.

Observe that in the case $k = 1$, the Strong Stable Set problem, we obtain a lower bound of $\Omega(n^{1/2-\epsilon})$ which is essentially tight in light of the upper bound given. The lower bound can be generalized for Set Packing to show that the $O(\sqrt{m})$ approximation in terms of the number of base elements is essentially the best possible.

We also obtain tight lower bounds for the Distance-t-IS problems defined earlier.

Theorem 3.10 *For any $\epsilon > 0$, the Distance-t-IS problem is hard to approximate within $n^{1-\epsilon}$ when t is odd, and within $n^{1/2-\epsilon}$ when t is even.*

Proof. First consider the odd case, $t = 2q+1$. Given a graph G, construct a graph H that contains a copy of G, a vertex u adjacent to every vertex of G, and a distinct path of q edges attached to each vertex of G. That is, $V(H) = \{v_i, w_{i,j} | v_i \in V(G), 1 \leq j \leq q\} \cup \{u\}$, and

$E(H) = E(G) \cup \{(u, v_i), (v_i, w_{i,1}), (w_{i,j}, w_{i,j+1}) | v_i \in V(G), 1 \leq j < q\}$. All pairs of vertices in the graph are of distance at most $2q + 2 = t + 1$. The only vertices of distance $t + 1$ are pairs $w_{i,q}, w_{j,q}$ of leaves on paths where (v_i, v_j) are non-adjacent. Hence, a Distance-t-IS in H is in one-to-one correspondence with an IS in G. Thus, the Distance-t-IS problem, for t odd, is no easier to approximate than the IS problem.

For the lower bound for the even case, we similarly append paths to each vertex of the construction for the Strong Stable Set problem. We invite the reader to verify the details. ∎

Acknowledgement

A comment by Hiroshi Nagamochi prompted us to greatly improve an early algorithm.

References

[1] R. B. Boppana and M.M. Halldórsson, Approximating maximum independent sets by excluding subgraphs, *BIT*, 32 (1992), 180–196.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability* (Freeman, New York, 1979).

[3] M.M. Halldórsson, Approximating the minimum maximal independence number, *Information Processing Letters* 46 (1993), 169–172.

[4] J. Håstad, Clique is hard to approximate within $n^{1-\epsilon}$, In *Proc. 37th IEEE Symp. on Found. of Comput. Sci.*, (1996), 627–636.

[5] J. Kratochvíl, Perfect codes in general graphs, monograph, Academia Praha (1991).

[6] J. Kratochvíl, P. Manuel and M. Miller, Generalized domination in chordal graphs, *Nordic Journal of Computing* 2 (1995), 41–50

[7] T.J. Schaefer, The complexity of satisfiability problems, In *Proc. 10th ACM Symp. on Theory of Computing* (1978), 216–226.

[8] J.A. Telle, Complexity of domination-type problems in graphs, *Nordic Journal of Computing* 1 (1994), 157–171.

[9] J.A. Telle and A. Proskurowski, Algorithms for Vertex Partitioning Problems on Partial k-Trees, *SIAM Journal on Discrete Mathematics 10–4*, to appear.