[Proceeding] Reoptimization of Some Maximum Weight Induced Hereditary Subgraph Problems

Reoptimization of maximum weight induced hereditary subgraph problems*

Nicolas Boria Jérôme Monnot Vangelis Th. Paschos[§]

LAMSADE, CNRS UMR 7243 and Université Paris-Dauphine

{boria,monnot,paschos}@lamsade.dauphine.fr

[§]Institut Universitaire de France

September 28, 2011

Abstract

The reoptimization issue studied in this paper can be described as follows: given an instance I of some problem Π, an optimal solution OPT for Π in I and an instance I' resulting from a local perturbation of I that consists of insertions or removals of a small number of data, we wish to use OPT in order to solve Π in I', either optimally or by guaranteeing an approximation ratio better than that guaranteed by an ex nihilo computation and with running time better that that needed for such a computation. We use this setting in order to study weighted versions of several representatives of a broad class of problems known in the literature as maximum induced hereditary subgraph problems. The main problems studied are MAX INDEPENDENT SET, MAX k-COLORABLE SUBGRAPH and MAX SPLIT SUBGRAPH under vertex insertions and deletions.

1 Introduction

Hereditary problems in graphs, also known as maximal subgraph problems, include a wide range of classical combinatorial optimization problems, such as MAX INDEPENDENT SET or MAX H-FREE SUBGRAPH. Most of these problems are known to be **NP**-hard, and even inapproximable within any constant approximation ratio unless **P** = **NP** [17, 20]. Some of them, and in particular MAX INDEPENDENT SET, have been intensively studied in the polynomial approximation framework [11, 15].

In what follows, we present approximation algorithms and inapproximability bounds for various hereditary problems in the reoptimization setting, which can be described as follows: considering an instance I of a given problem Π with a known optimum OPT, and an instance I' which results from a local perturbation of I, can the information provided by OPT be used to solve I' in a more efficient way (i.e., with a lower complexity and/or with a better approximation ratio) than if this information wasn't available?

The reoptimization setting was introduced in [1] for METRIC TSP. Since then, many other optimization problems were discussed in this setting, including STEINER TREE [5, 8, 9, 14], MINIMUM SPANNING TREE [13], as well as various versions of TSP [4, 7, 10]. In all cases, the goal is to propose reoptimization algorithm that outperform their deterministic counterparts in terms of complexity and/or approximation ratio. In [6], the MAX INDEPENDENT SET problem, as well as MIN VERTEX COVER and MIN SET COVER problems, are discussed in a similar setting up to

*Research supported by the French Agency for Research under the DEFIS program TODO, ANR-09-EMER-010

the fact that perturbations there concerned the edge-set of the initial graph. The authors of [6] manage to provide optimal approximation results under the basic assumption that the initial solution is not necessarily optimal but ρ-approximate.

When one deals with hereditary problems, and I' results from a perturbation of the vertex set (insertion or deletion), solutions of I remain feasible in I'. This property is very interesting when reoptimizing hereditary problems, and makes most of them APX in the reoptimization setting. For exemple, a very simple algorithm provides a (1/2)-approximation for a whole class of hereditary problems when a single vertex is inserted [3]. In what follows, we improve on this result by presenting algorithms designed for four specific hereditary problems, and also provide inapproximability bounds. We also discuss the reoptimization setting where vertices are deleted, which, as we will see, is much harder to approximate.

The paper is organized as follows: general properties regarding hereditary problems are presented in Section 2, while Sections 3 and 4 present approximation and inapproximability results regarding respectively vertex insertion and deletion. In Table 1, our main results are presented. One can see there that upper bounds match lower bounds everywhere.

Hereditary problems	Approximation ratios	Inapproximability bounds
MAX k-COLORABLE SUBGRAPH (insertion of h vertices)	$\max\left\{\frac{k}{k+h}, \frac{1}{2}\right\}$	$\max\left\{\frac{k}{k+h}, \frac{1}{2}\right\} + \varepsilon$
MAX k-COLORABLE SUBGRAPH (deletion of $h < k$ vertices)	$\frac{k-h}{k}$	$\frac{k-h}{k} + \varepsilon$
MAX k-COLORABLE SUBGRAPH (deletion of $h \geqslant k$ vertices)	$\emptyset$	$n^{-\varepsilon}$
MAX SPLIT SUBGRAPH (insertion of $h \leqslant 2$ vertices)	$\frac{h+1}{2h+1}$	$\frac{h+1}{2h+1} + \varepsilon$
MAX SPLIT SUBGRAPH (insertion of $h = 3$ vertices)	$\frac{5}{9}$	$\frac{5}{9} + \varepsilon$
MAX SPLIT SUBGRAPH (insertion of $h \geqslant 4$ vertices)	$\frac{1}{2}$	$\frac{1}{2} + \varepsilon$

Table 1: Summary of the results

This paper is part of a larger work [12] devoted to the study of five maximum weight induced hereditary subgraph problems, namely, MAX INDEPENDENT SET, MAX k-COLORABLE SUBGRAPH, MAX P_k-FREE SUBGRAPH, MAX SPLIT SUBGRAPH and MAX PLANAR SUBGRAPH. For reasons of length limits some of the results are given without detailed proofs that can be found in appendix.

2 Preliminaries

Before presenting properties and results regarding reoptimization problems, we will first give formal definitions of what are reoptimization problems, reoptimization instances, and approximate reoptimization algorithms:

Definition 1. An optimization problem Π is given by a quadruple $(\mathcal{I}_\Pi, Sol_\Pi, m_\Pi, goal(\Pi))$ where: $\mathcal{I}_\Pi$ is the set of instances of Π; given $I \in \mathcal{I}_\Pi$, $Sol_\Pi(I)$ is the set of feasible solutions of I; given $I \in \mathcal{I}_\Pi$, and $S \in Sol_\Pi(I)$, $m_\Pi(I, S)$ denotes the value of the solution S of the instance I; $goal(\Pi) \in \{\min, \max\}$. ∎

A reoptimization problem $R\Pi$ is given by a pair $(\Pi, R_{R\Pi})$ where: Π is an optimization problem as defined in Definition 1; $R_{R\Pi}$ is a rule of modification on instances of Π, such as addition, deletion or alteration of a given amount of data; given $I \in \mathcal{I}_\Pi$ and $R_{R\Pi}$, $modif_{R\Pi}(I, R_{R\Pi})$ denotes the set of instances resulting from applying modification $R_{R\Pi}$ to I; notice that $modif_{R\Pi}(I, R_{R\Pi}) \subset \mathcal{I}_\Pi$.

For a given reoptimization problem $R\Pi(\Pi, R_{R\Pi})$, a reoptimization instance $I_{R\Pi}$ of $R\Pi$ is given by a triplet (I, S, I'), where: I denotes an instance of Π, referred to as the *initial* instance; S denotes a feasible solution for Π on the initial instance I; I' denotes an instance of Π in $modif_{R\Pi}(I, R_{R\Pi})$; I' is referred to as the *perturbed* instance. For a given instance $I_{R\Pi}(I, S, I')$ of $R\Pi$, the set of feasible solutions is $Sol_\Pi(I')$.

Definition 2. For a given optimization problem $R\Pi(\Pi, R_{R\Pi})$, a reoptimization algorithm A is said to be a ρ-approximation reoptimization algorithm for $R\Pi$ if and only if: (i) A returns a feasible solution on all instances $I_{R\Pi}(I, S, I')$; (ii) A returns a ρ-approximate solution on all reoptimization instances $I_{R\Pi}(I, S, I')$ where S is an optimal solution for I. ∎

Note that Definition 2 is the most classical definition found in the literature, as well as the one used in this paper. However, an alternate (and more general) definition exists (used for example in [5, 6, 8, 9]), where a ρ_1-approximation reoptimization algorithm for $R\Pi$ is supposed to ensure a $\rho_1\rho_2$ approximation on any reoptimization instance $I_{R\Pi}(I, S, I')$ where S is a ρ_2 approximate solution in the initial instance I.

A property $\mathcal{P}$ on a graph is hereditary if the following holds: if the graph satisfies $\mathcal{P}$, then $\mathcal{P}$ is also satisfied by all its induced subgraphs. Following this definition, independence, planarity, bipartiteness are three examples of hereditary properties: in a given graph, any subset of an independent set is an independent set itself, and the same holds for planar and bipartite subgraphs. On the opposite hand, connectivity is no hereditary property since there might exist some subsets of G whose removal disconnect the graph. It is also well known that any hereditary property in graphs can be characterized by a set of forbidden subgraphs or minors [18].

In other words a property $\mathcal{P}$ is hereditary if and only if, there is a set of graphs H such that every graph that verifies $\mathcal{P}$ does not admit any graph in H as a minor or as an induced subgraph. To revisit the three examples of hereditary properties presented before: an independent set is characterized by one forbidden subgraph, a K_2 (a clique on 2 vertices, i.e., an edge); a planar graph is characterized by two forbidden minors: K_5 (a clique on 5 vertices), and $K_{3,3}$ (a complete bipartite graph with both its color-classes of size 3) (this result is known as Wagner's Theorem [19]); a bipartite graph is characterized by a infinite set of forbidden induced subgraphs: all odd cycles $H = \{C_{2n+1}, n \geqslant 1\}$.

Definition 3. Let $G(V, E, w)$ be a vertex-weighted graph with $w(v) \geqslant 0$, for any $v \in V$. The *max weighted induced subgraph with property* $\mathcal{P}$ problem (or, for short, *max weighted subgraph problem*) is the problem consisting, given a graph $G(V, E)$, of finding a subset of vertices S such that $G[S]$ satisfies a given property $\mathcal{P}$ and maximizes $w(S) = \sum_{v \in S} w(v)$. We call hereditary problems all such problems where $\mathcal{P}$ is a hereditary property. ∎

For instance, MAX WEIGHTED INDEPENDENT SET, MAX WEIGHTED INDUCED BIPARTITE SUBGRAPH, MAX WEIGHTED INDUCED PLANAR SUBGRAPH are three classical hereditary problems that correspond to the three hereditary properties as defined in Definition 3.

As it is proved in [17] (see Theorem 1 just below) most hereditary problems are highly inapproximable unless $\mathbf{P} = \mathbf{NP}$.

Theorem 1. ([17]) *There exists an $\varepsilon \in (0, 1)$ such that the maximum subgraph problem cannot be approximated with ratio $n^{-\varepsilon}$ in polynomial time for any nontrivial hereditary property that is false for some clique or independent set, or more generally is false for some complete multipartite graph, unless* $\mathbf{P} = \mathbf{NP}$.

Throughout the paper, all inapproximability results will be obtained by the same technique, which we sketch out here.

Considering an unweighted graph $H(V,E)$ on which one wants to solve a given hereditary problem Π, known to be inapproximable within any constant ratio, we build a reoptimization instance I_p, where p denotes a vector of fixed size (i.e., independent of the size n of G; so, $|p|$ is a fixed constant) that contains integer parameters between 1 and n. This instance is characterized by an initial graph G_p (that contains H), with a known solution, and a perturbed instance G'_p.

Then, we prove that, for some specific (yet unknown) value p' of the parameter vector p, an optimal solution can be easily determined in the initial graph $G_{p'}$, and a ρ-approximate solution $S_{p'}$ in $G'_{p'}$ necessarily induces a solution $S_{p'}[V]$ in H, that is a constant approximation for the initial problem in $H(V,E)$.

Considering that the vector p can take at most $n^{|p|}$ possible values, it is possible in polynomial time to build all instances I_p, to run the polynomial ρ-approximation algorithm on all of them, and to return the best set $S_{p^*}[V]$ as solution for Π in H. The whole procedure is polynomial and ensures a constant-approximation for Π, which is impossible unless $\mathbf{P} = \mathbf{NP}$, so that a ρ-approximation algorithm cannot exist for the considered reoptimization version of Π, unless $\mathbf{P} = \mathbf{NP}$.

In the sequel, G_p and G'_p will denote initial and perturbed instances, while OPT_p and OPT'_p will denote optimal solutions in G_p and G'_p, respectively. For simplicity and when no confusion arises, we will omit subscript p. The function w refers to the weight function, taking a vertex, a vertex set, or a graph as input (the weight of a graph is defined as the sum of weights of its vertices). Finally, note that throughout the whole paper, the term "subgraph" will always implicitly refer to "induced subgraph".

We conclude this section by the following emphasized remark. *The proofs of all of our results in this paper always work, even if we assume that a ρ-approximate solution is given instead of an optimal one. In this case, the bounds claimed are simply multiplied by ρ.*

3 Vertex insertion

Under vertex insertion, the inapproximability bound of Theorems 1 is easily broken. In [3], a very simple strategy, denoted by `R1` in what follows, provides a $(1/2)$-approximation for any hereditary problem. This strategy consists of outputting the best solution among the newly inserted vertex and the initial optimum. Moreover, this strategy can also be applied when a constant number h of vertices is inserted: it suffices to output the best solution between an optimum in the h newly inserted vertices (that can be found in $O(2^h)$ through exhaustive search) and the initial optimum. The $1/2$ approximation ratio is also ensured in this case [3].

Note that an algorithm similar to `R1` was proposed for KNAPSACK in [2]. Indeed, this problem, although not being a graph problem, is hereditary in the sense defined above, so that returning the best solution between a newly inserted item and the initial optimum ensures a $(1/2)$-approximation ratio. The authors also show that any reoptimization algorithm that does not consider objects discarded by the initial optimal solution cannot have ratio better than $1/2$.

In what follows, we start by proving that this approximation ratio is the best constant approximation ratio one can achieve for the MAX INDEPENDENT SET problem (Section 3.1), unless $\mathbf{P} = \mathbf{NP}$. Then, we present other simple polynomial constant-approximation strategies, as well as inapproximability bounds for MAX k-COLORABLE SUBGRAPH and MAX SPLIT SUBGRAPH.

3.1 MAX INDEPENDENT SET

Since MAX INDEPENDENT SET is a hereditary problem, strategy `R1` provides a simple and fast $(1/2)$-approximation in the reoptimization setting under insertion of one vertex. We will now prove that this ratio is the best one can hope, unless $\mathbf{P} = \mathbf{NP}$.

Proposition 1. *In the reoptimization setting, under one vertex insertion,* MAX INDEPENDENT SET *is inapproximable within ratio* $1/2 + \varepsilon$ *in polynomial time, unless* $\mathbf{P} = \mathbf{NP}$.

Proof. By contradiction, assume that there exists a reoptimization approximation algorithm `A` for MAX INDEPENDENT SET, which, in polynomial time, computes a solution with approximation ratio bounded by $1/2 + \varepsilon$. Now, consider a graph $H(V, E)$. All n vertices in V have weight 1, and no assumption is made on V. Note that in such a graph (which is actually unweighted), MAX INDEPENDENT SET is inapproximable within any constant ratio, unless $\mathbf{P} = \mathbf{NP}$.

We will now make use of `A` to build an ε-approximation for MAX INDEPENDENT SET in H, and thus prove that such an algorithm cannot exist. Denote by α the independence number associated with H, that is, the - unknown - cardinality of an optimal independent set in H, and consider the following instance I_α of MAX INDEPENDENT SET in the reoptimization setting (here the vector p is an 1-vector, so it is an integer between 1 and n):

• The initial graph denoted $G_\alpha(V_\alpha, E_\alpha)$ is obtained by adding a single vertex x to V, with weight α, and connecting this new vertex to every vertex in V. Thus, $V_\alpha = V \cup \{x\}$, and $E_\alpha = E \cup \bigcup_{v_i \in V}(x, v_i)$. In this graph, a trivial optimum independent set is $\{x\}$. This trivial solution will be the initial optimum used in the reoptimization instance.

• The perturbed graph $G'_\alpha(V'_\alpha, E'_\alpha)$ is obtained by adding a single vertex y to G_α, also with weight α, and connecting this new vertex to vertex x only.

Denote by OPT' an optimal independent set in G'_α. Notice that y (whose weight is α) can be added to an optimal independent set in H (whose weight is also α) to produce a feasible solution in G'_α, so that: $w(\mathrm{OPT}') \geqslant 2\alpha$.

Now, suppose that one runs the approximation algorithm `A` on the so-obtained reoptimization instance I_α. By hypothesis on `A`, it holds that $w(S_\alpha) \geqslant (1/2 + \varepsilon)w(\mathrm{OPT}') \geqslant (1 + \varepsilon)\alpha$.

Considering the lower bound on its weight, we can assert that the solution returned by `A`, does not contain x (the only independent set containing x is x itself, and thus it cannot have weight more than α). Moreover, it must contain y, otherwise it would be restricted to an independent set in G, so it couldn't have weight more than α. So, it holds that $w(S_\alpha[V]) = w(S_\alpha) - w(y) \geqslant (1 + \varepsilon)\alpha - \alpha = \varepsilon\alpha$, where $w(S_\alpha[V])$ denotes the restriction of S_α to the initial graph H.

Now, consider the following approximation algorithm `A1` for MAX INDEPENDENT SET:

> build n reoptimization instances I_i in the same way as I_α (only the weights of vertices x and y will be different from one instance to the other), for $i = 1, \ldots, n$, and run the reoptimization `A` on each of them. Denoting by S_i the solution returned by `A` on instance I_i, and $S_i[V]$ its restriction to the initial graph H, output the set $S_{\max}[V]$ with maximal weight among $S_i[V]$'s.

Obviously, considering that $1 \leqslant \alpha \leqslant n$, it holds that $S_{\max}[V] \geqslant S_\alpha[V] \geqslant \varepsilon\alpha$. Thus, algorithm `A1`, using n times the algorithm `A` as subroutine, produces in polynomial time an ε-approximation for (unweighted) MAX INDEPENDENT SET, which is impossible unless $\mathbf{P} = \mathbf{NP}$. ∎

Note that the results also hold when a constant number h of vertices are inserted. Indeed, it is easy to see that all the arguments of the proof remain valid when the set of inserted vertices is $\{y_1, \ldots, y_h\}$ each with weight α/h and connected only to vertex x.

Proposition 2. *Under insertion of one vertex and unless* $\mathbf{P} = \mathbf{NP}$, MAX INDEPENDENT SET *is not approximable within ratio* $(1/2 + (1/(n-1)^\epsilon)$*, for any* $\epsilon > 0$*, where* n *is the order of the perturbed graph.*

Let us note that inapproximability bounds stated in Propositions 3, 6 and 8, that are of the form $\rho + \varepsilon$, $\varepsilon \in (0, 1)$, can be strengthened to $\rho + n^{-\varepsilon}$. Indeed, the proofs of these propositions are based upon the argument that the existence of a $(\rho+\varepsilon)$-approximation algorithm for a given reoptimization problem $R\Pi$ induce the existence of a $O(\varepsilon)$-approximation algorithm for the "static" support Π of $R\Pi$. However, the "static" problems dealt in these propositions are not only inapproximable within $O(\varepsilon)$, unless $\mathbf{P}=\mathbf{NP}$, but within $O(n^{-\varepsilon})$. Hence, revisiting their proofs, one can replace ε by $n^{-\varepsilon}$ getting so inapproximability bounds $\rho+n^{-\varepsilon}$ instead.

3.2 MAX k-COLORABLE SUBGRAPH

Given a graph $G(V,E,w)$ and a constant $k\leqslant n$, the MAX k-COLORABLE SUBGRAPH problem consists of determining the maximum-weight subset $V'\subseteq V$ that induces a subgraph of G that is k-colorable.

Using the same technique, the result of Section 3.1 can be generalized to the MAX k-COLORABLE SUBGRAPH problem as shows the following proposition the proof of which can be found in Appendix A.

Proposition 3. *In the reoptimization setting, under the insertion of h vertices,* MAX k-COLORABLE SUBGRAPH *is inapproximable within ratio* $\max\left\{\frac{k}{k+h},\frac{1}{2}\right\}+\varepsilon$ *in polynomial time, unless* $\mathbf{P}=\mathbf{NP}$.

This inapproximability bound is tight for the MAX INDEPENDENT SET problem (which can also be defined as the MAX 1-COLORABLE SUBGRAPH), where an easy reoptimization algorithm produces solutions with approximation ratio bounded by 1/2. We now show that this tightness holds also for MAX k-COLORABLE SUBGRAPH for any $k\geqslant 1$.

Proposition 4. *Under the insertion of h vertices,* MAX k-COLORABLE SUBGRAPH *problem is* $\left(\max\left\{\frac{k}{k+h},\frac{1}{2}\right\}\right)$*-approximable.*

Proof. Consider a reoptimization instance I of the MAX k-COLORABLE SUBGRAPH problem. The initial graph is denoted by $G(V,E)$, and the perturbed one by $G'(V',E')$ where $V'=V\cup\{Y\}$, $Y=y_1,\ldots,y_h$. Let OPT and OPT′ denote optimal k-colorable graphs on G, and G' respectively. The initial optimum OPT is given by a set of k independent sets: $(S_1,\ldots,S_k)$, and w.l.o.g., suppose $w(S_1)\geqslant w(S_2)\geqslant\ldots,\geqslant w(S_k)$. Now, consider the following algorithm:

> if $h\geqslant k$, then apply the algorithm R1, described in [3] (ensuring a 1/2-approximate solution for any hereditary problem), else ($h<k$), let $\mathrm{SOL}_1=\left(\bigcup_{i=1}^{k-h}S_i\right)\cup\{Y\}$, and $\mathrm{SOL}_2=\mathrm{OPT}$; return the best solution SOL between SOL_1 and SOL_2.

First, considering that the restriction of OPT′ to V cannot define a better solution than OPT, $w(\mathrm{SOL}_2)=w(\mathrm{OPT})\geqslant w(\mathrm{OPT}')-w(Y)$. Note that SOL_1 is a feasible solution. Indeed, $\bigcup_{i=1}^{k-h}S_i$ induces a $(k-h)$-colorable subgraph, thus, adding h vertices to it (here, the set Y) induces a k-colorable subgraph. Moreover, $w\left(\bigcup_{i=1}^{k-h}S_i\right)\geqslant\frac{k-h}{k}w(\mathrm{OPT})\geqslant\frac{k-h}{k}\left(w(\mathrm{OPT}')-w(Y)\right)$; so, $w(\mathrm{SOL}_1)\geqslant\frac{k-h}{k}\left(w(\mathrm{OPT}')-w(Y)\right)+w(Y)\geqslant\frac{k-h}{k}w(\mathrm{OPT}')+\frac{h}{k}w(Y)$.

Summing expressions for $w(\mathrm{SOL}_1)$ and $w(\mathrm{SOL}_2)$ given just above with coefficients 1 and k/h, respectively, one gets $w(\mathrm{SOL}_2)+\frac{k}{h}w(\mathrm{SOL}_1)\geqslant\frac{k}{h}w(\mathrm{OPT}')$

Taking into account that $\frac{k+h}{h}w(\mathrm{SOL})\geqslant w(\mathrm{SOL}_2)+\frac{k}{h}w(\mathrm{SOL}_1)$, it holds that $w(\mathrm{SOL})\geqslant\frac{k}{k+h}w(\mathrm{OPT}')$, and the proof is completed. ∎

3.3 MAX SPLIT SUBGRAPH

Given a graph $G(V,E,w)$, the MAX SPLIT SUBGRAPH problem consists of determining a maximum-weight subset $V'\subseteq V$ that induces a split subgraph of G. A split graph is a graph whose vertices can be partitioned into two sets C and S, C being a clique, and S being an independent set. Any subset of a clique remains a clique, and any subset of independent set remains an independent set, hence, being a split graph is a hereditary property. Moreover, considering that the property is false for a complete bipartite graph with at least two vertices in each independent set, the result of Theorem 1 applies to the MAX SPLIT SUBGRAPH problem. So MAX SPLIT SUBGRAPH is inapproximable within any constant ratio, unless $\mathbf{P} = \mathbf{NP}$.

We prove that this strong inapproximability result does not hold in the reoptimization setting, but we first need to prove the following lemma, the proof of which can be found in Appendix B.

Lemma 1. *Let G be a graph with $h \leqslant 3$ vertices. It holds that $w(G_S) + w(G_C) \geqslant \frac{h+1}{h} w(G)$ if $h \leqslant 2$ and $w(G_S) + w(G_C) \geqslant \frac{5}{4} w(G)$ if $h = 3$, where G_S and G_C respectively denote an optimal independent set and an optimal clique in G.*

Proposition 5. *Under insertion of h vertices,* MAX SPLIT SUBGRAPH *problem is $\frac{h+1}{2h+1}$-approximable for $h \leqslant 2$, and $\frac{5}{9}$-approximable for $h = 3$.*

Proof. Consider a reoptimization instance I of the MAX SPLIT SUBGRAPH problem. The initial graph is denoted by $G(V, E)$, and the perturbed one $G'(V', E')$, where $V' = V \cup Y$ where $|Y| = h \leqslant 3$. Let OPT and OPT$'$ denote optimal split-graphs on G, and G' respectively. The initial optimum OPT is given by a clique C and an independent set S. Let Y_S and Y_C denote optimal independent sets and cliques in Y. Consider the following algorithm:

> let $\mathrm{SOL}_1 = S \cup Y_C$, $\mathrm{SOL}_2 = C \cup Y_S$, and $\mathrm{SOL}_3 = \mathrm{OPT}$; return the best solution SOL among SOL_1, SOL_2, and SOL_3.

First, noticing that $S \cup Y_C$ and $C \cup Y_S$ both define split graphs, it holds that the algorithm returns a feasible solution. Then summing $w(\mathrm{SOL}_1)$, and $w(\mathrm{SOL}_2)$, we get the following equality:

$$w(\mathrm{SOL}_1) + w(\mathrm{SOL}_2) = w(C) + w(S) + w(Y_C) + w(Y_S) \geqslant \begin{cases} w(\mathrm{OPT}) + \frac{h+1}{h} w(Y) & \text{if } h \leqslant 2, \\ w(\mathrm{OPT}) + \frac{5}{4} w(Y) & \text{if } h = 3. \end{cases}$$

The second line follows noticing that $w(C) + w(S) = w(\mathrm{OPT})$, and taking into account that, according to Lemma 1, $w(Y_S) + w(Y_C) \geqslant \frac{h+1}{h} w(Y)$ if $h \leqslant 2$, and $w(Y_S) + w(Y_C) \geqslant \frac{5}{4} w(Y)$ if $h = 3$. Notice that, since $w(\mathrm{OPT}) \geqslant w(\mathrm{OPT}') - w(Y)$, it holds that:

$$\begin{aligned} w(\mathrm{SOL}_1) + w(\mathrm{SOL}_2) &\geqslant \begin{cases} w(\mathrm{OPT}') + \frac{1}{h} w(Y) & \text{if } h \leqslant 2 \\ w(\mathrm{OPT}') + \frac{1}{4} w(Y) & \text{if } h = 3 \end{cases} & (1) \\ w(\mathrm{SOL}_3) &\geqslant w(\mathrm{OPT}') - w(Y) & (2) \end{aligned}$$

Finally, summing (1) and (2) with coefficients h and 1, if $h \leqslant 2$, and 4 and 1 if $h = 3$:

$$\begin{cases} (2h+1) w(\mathrm{SOL}) \geqslant h(w(\mathrm{SOL}_1) + w(\mathrm{SOL}_2)) + w(\mathrm{SOL}_3) \geqslant (h+1) w(\mathrm{OPT}') & \text{if } h \leqslant 2 \\ 9 w(\mathrm{SOL}) \geqslant 4(w(\mathrm{SOL}_1) + w(\mathrm{SOL}_2)) + w(\mathrm{SOL}_3) \geqslant 5 w(\mathrm{OPT}') & \text{if } h = 3 \end{cases}$$

and the proof is completed. ∎

Recall that for any h (and a fortiori for $h \geqslant 4$) the problem is 1/2-approximable by the algorithm `R1` presented in [3]. We prove that these simple approximation algorithms achieve the best constant ratios possible.

Proposition 6. *Under vertex insertion,* MAX SPLIT SUBGRAPH *is inapproximable within ratios $\frac{h+1}{2h+1} + \varepsilon$ when $h \leqslant 2$, $\frac{5}{9} + \varepsilon$ when $h = 3$, and $\frac{1}{2} + \varepsilon$ when $h \geqslant 4$ in polynomial time, unless* $\mathbf{P} = \mathbf{NP}$.

Proof. [*Sketch*] Consider an unweighted graph H where one wishes to solve MAX SPLIT SUBGRAPH and denote by α its independence number, β its clique number. Construct the initial instance H_{α,β,h_1,h_2} ($h_1, h_2 \geqslant 1$) as described in Appendix C.1.

Assume $h \leqslant 2$. We build a reoptimization instance, $I_{\alpha,\beta,h}$ in the following way:

• The initial graph is the graph $G_{\alpha,\beta,h,1}$. We prove in Appendix C.1 that X is an optimum on this graph. Here, its weight is $(h+1)\gamma + 2$.

• The perturbed graph $G'_{\alpha,\beta,h,1}$ is obtained by adding a set of vertices Y to $G_{\alpha,\beta,h,1}$, which consists of an independent set of h vertices, each with weight γ. All vertices in Y are connected to all vertices in X_S only.

The overall structure is represented in Figure 1, as well as the weight of optimal independent sets and cliques (denoted by S^* and C^* in the Figure) in all sets V, X_C, X_S and Y.

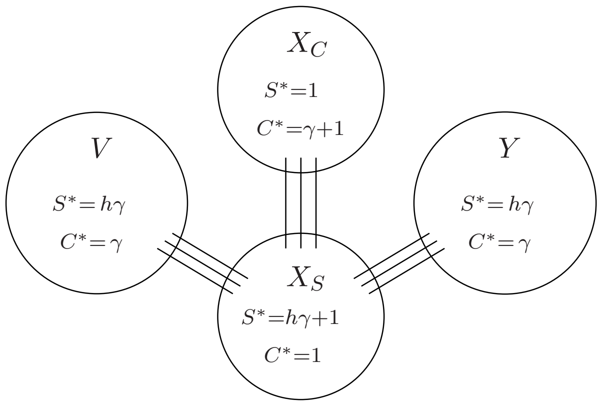


Figure 1: The reoptimization instance $I_{\alpha,\beta,h}$, $h \leqslant 2$

Notice that, in the perturbed graph $Y \cup X_C \cup F^*$ (where F^* is an optimal independent set in V) defines an split graph of weight $(2h+1)\gamma + 1$. Indeed, $Y \cup F^*$ defines an independent set, while X_C defines an clique. Thus, denoting by OPT' an optimal split graph in $G'_{\alpha,\beta,h,1}$, it holds that $w(\mathrm{OPT}') \geqslant (2h+1)\gamma + 1$.

Suppose that, for a given $h \leqslant 2$, there exists an approximation algorithm `A` for the reoptimization version of MAX SPLIT SUBGRAPH, which provides an approximation ratio bounded by $\frac{h+1}{2h+1} + \varepsilon$, under the insertion of h vertices. Denoting by $S_{\alpha,\beta,h}$ a solution returned by this algorithm on the reoptimization instance $I_{\alpha,\beta,h}$ we just described, it holds that:

$$w(S_{\alpha,\beta,h}) \geqslant \left(\frac{h+1}{2h+1} + \varepsilon\right) \mathrm{OPT}' \geqslant (1+\varepsilon)(h+1)\gamma$$

However, a split graph SG in $X \cup Y$ (and a fortiori the restriction of $S_{\alpha,\beta,h}$ to $X \cup Y$, denoted by $S_{\alpha,\beta,h}[X \cup Y]$) cannot have weight more than $(h+1)\gamma + 2$. We distinguish here the following two cases.

Case 1. SG takes at most one vertex in X_S. Then $w(SG[X_S]) \leqslant 1$, and thus:

$$w(SG) = w(SG[X_S]) + w(X_C) + w(Y) \leqslant 1 + \gamma + 1 + h\gamma = (h+1)\gamma + 2$$

Case 2. SG takes at least two vertices in X_S, then the independent set in SG can contain only vertices of X_S. In other words, the vertices of $Y \cup X_C$ can only be part of the clique in SG. It is quite obvious that the biggest clique in $Y \cup X_C$ is X_C itself so that in this case $w(SG) \leqslant w(X) = (h+1)\gamma + 2$. One immediately derives from this result that $w(S_{\alpha,\beta,h}[X \cup Y]) \leqslant (h+1)\gamma + 2$ and Case 2 is concluded.

So, in both cases it holds that $w(S_{\alpha,\beta,h}[V]) = w(S_{\alpha,\beta,h}) - w(S_{\alpha,\beta,h}[X \cup Y]) \geqslant \varepsilon\gamma - 2$. Considering that γ is not a constant, if an algorithm A exists, one can get in polynomial time a constant-approximate solution for MAX SPLIT SUBGRAPH in the graph $H_{\alpha,\beta,h,1}$, which is impossible unless $\mathbf{P} = \mathbf{NP}$.

The cases $h = 3$ and $h \geqslant 4$ are discussed in Appendices C.3 and C.4, respectively. ∎

4 Vertex deletion

Let us consider now the opposite kind of perturbations: vertex-deletion. When dealing with hereditary optimization problems, some properties discussed just above still remain valid, while some others do not. As before, let us consider a given instance of a hereditary problem, for which we know an optimal solution OPT. Consider now that one vertex of the graph is deleted, along with its incident edges. Two cases might occur:

(i) the deleted vertex y was not part of the initial optimum, so it remains the same in the new graph.

(ii) y was part of the initial optimum, and might even have been one of its most important elements. Though having a priori no information on the quality of the initial optimum $\mathrm{OPT} \setminus \{y\}$ in the new graph G' (or rather what is left of it), we can still assert that $\mathrm{OPT} \setminus \{y\}$ remains a feasible solution in the new graph.

In what follows, we discuss to what extent the techniques used in the case of insertion can be applied to the case of deletion. As in Section 3, we will start by an inapproximability result on all inapproximable hereditary problems and we provide some tight positive results for MAX k-COLORABLE SUBGRAPH. We finally present general techniques for reoptimizing hereditary problems in graphs of bounded degree.

4.1 A general negative result and some applications

When dealing with MAX INDEPENDENT SET, the whole initial optimum can disappear when deleting a single vertex, since *the minimal size of a maximal solution is 1*, put differently, a single vertex can be a maximal solution. However, this fact does not hold for any hereditary property. Consider for example the MAX BIPARTITE SUBGRAPH problem. Regarding this problem, a single vertex cannot define a maximal solution, and it takes at least two deleted vertices to delete the whole initial optimum. We derive from this idea the following general inapproximability result:

Proposition 7. *Let $M(\Pi)$ denote the minimal size of a maximal solution for a given hereditary problem Π. Under the deletion of $h \geqslant M(\Pi)$ vertices, Π is inapproximable within any ratio $n^{-\varepsilon}$ in polynomial time, unless $\mathbf{P} = \mathbf{NP}$.*

Proof. Consider an instance of a given unweighted non trivial hereditary problem Π, that consists of a graph $H(V, E)$. We build the following reoptimization instance I: The initial graph G is obtained by adding to H a set of vertices Y of size $h \geqslant M(\Pi)$. This set contains a gadget of size $M(\Pi)$ that constitutes a maximal solution in G, where each vertex has weight n, and $h - M(\Pi)$ vertices with weight 0 (which will be ignored in what follows). The perturbed graph is the graph H.

It is clear that the $M(\Pi)$ vertices of weight n in Y define an optimal solution in the initial graph G: This gadget is feasible and maximal, so that in G an optimal solution has weight at least $M(\Pi)n$. On the other hand, any solution that does not take the whole gadget has weight at most $(M(\Pi) - 1)n + \mathrm{OPT} \leqslant M(\Pi)n$, where OPT denotes the cardinality of an optimal solution in H. Thus, Y can be considered as the initial optimum of the reoptimization instance I.

Consider a reoptimization algorithm A, which, for a given $h \geqslant M(\Pi)$, does provide an approximation ratio $n^{-\varepsilon}$ under the deletion of h vertices. When using it on the reoptimization instance I, we just described, this algorithm produces a $n^{-\varepsilon}$-approximate solution in H in polynomial time, which is impossible unless $\mathbf{P} = \mathbf{NP}$. ∎

Corollary 1. MAX k-COLORABLE SUBGRAPH, *under deletion of* $h \geqslant k$ *vertices, and* MAX SPLIT SUBGRAPH, *under deletion of* $h \geqslant 3$ *vertices, are inapproximable within ratio* $n^{-\varepsilon}$ *unless* $\mathbf{P} = \mathbf{NP}$.

For MAX k-COLORABLE SUBGRAPH and MAX SPLIT SUBGRAPH, it suffices to notice that K_k's can define maximal solution for both these problems. For MAX SPLIT SUBGRAPH, notice that three vertices can define a maximal solution for MAX SPLIT SUBGRAPH: revisiting the proof of Proposition 7, build the gadget in Y as follows: two vertices y_1, y_2 that are connected only one to the other, and a vertex y_3 connected to all vertices in H. Clearly, $\{y_1, y_2, y_3\}$ defines a maximal (and optimal) solution in G.

Following Corollary 1, it holds that no constant approximation ratio can be expected in polynomial when more than k vertices are deleted, However if the number of deleted vertices is smaller than k, the non deleted part of the initial optimum is non-empty. Following this idea we provide the following positive result for MAX k-COLORABLE SUBGRAPH. Its proof is in Appendix D.

Proposition 8. *Under deletion of* $h < k$ *vertices,* MAX k-COLORABLE SUBGRAPH *is approximable within ratio* $\frac{k-h}{k}$.

This constant approximation ratio is the best one can obtain by a polynomial algorithm (unless $\mathbf{P} = \mathbf{NP}$) as the following proposition claims (see Appendix E for its proof).

Proposition 9. *Under deletion of* $h < k$ *vertices,* MAX k-COLORABLE SUBGRAPH *is inapproximable within ratio* $\frac{k-h}{k} + \varepsilon$ *in polynomial time, unless* $\mathbf{P} = \mathbf{NP}$.

4.2 Restriction to graphs of bounded degree

We will start with a general result that applies to any hereditary problem which can be characterized in terms of forbidden subgraphs of bounded diameter. We denote such problem by MAX H-FREE SUBGRAPH problem. Then we provide an example of what this general result amounts to regarding the MAX INDEPENDENT SET problem under vertex deletion in graphs of bounded degree.

Proposition 10. *In graphs of degree bounded by* Δ*, reoptimization of* MAX H-FREE SUBGRAPH *(where each forbidden subgraph has diameter bounded by* d*) under deletion of a constant number* h *of vertices is equivalent to reoptimization of the same problem under the insertion of* $h\Delta^d$ *vertices.*

Proof. Consider a reoptimization instance I of MAX H-FREE SUBGRAPH given by an initial graph $G(V, E)$ with degree bounded by Δ, and with a known optimal solution OPT, and a perturbed graph $G'(V', E') = G[V \setminus Y]$, $|Y| = h$.

Recall that all forbidden subgraphs have diameter bounded by a constant d. Let FS (for forbidden subgraph) denote the set of vertices that are reachable from a deleted vertex by a path that has at most d edges. Obviously $|FS| \leqslant h\Delta^d$. It holds that $\mathrm{OPT} \setminus Y$ is an optimal solution on $G'[V' \setminus (FS \setminus \mathrm{OPT})]$.

Indeed, consider a feasible solution S on the graph $G'[V' \setminus (FS \setminus \mathrm{OPT})]$ each vertex of this graph is either not reachable from any deleted vertex by a path of length d, thus it cannot be part of a forbidden subgraph in G along with vertices of $\mathrm{OPT} \cap Y$, or it is in OPT; considering that OPT is a feasible solution in G, these vertices cannot form a forbidden subgraph in G along with $\mathrm{OPT} \cap Y$.

In all, no vertex in S can form a forbidden subgraph along with $\mathrm{OPT} \cap Y$, so that $S \cup (\mathrm{OPT} \cap Y)$ is necessarily a feasible solution in G. Now, suppose that $w(S) > w(\mathrm{OPT} \setminus Y)$. This induces that $w(S \cup (\mathrm{OPT} \cap Y)) > w(\mathrm{OPT})$, which is impossible considering that $S \cup (\mathrm{OPT} \cap Y)$ is feasible in G. We proved that $\mathrm{OPT} \setminus Y$ is an optimal solution on $G'[V' \setminus (FS \setminus \mathrm{OPT})]$.

Hence, any reoptimization instance I of MAX H-FREE SUBGRAPH under deletion of h vertices can be characterized by a graph $G''(V'', E'') = G'[V' \setminus (FS \setminus \mathrm{OPT})]$ with a known optimal solution $\mathrm{OPT} \setminus Y$, and a graph $G'(V', E')$ where one wants to optimize the problem. The graph G' contains G'' as a subgraph, and has at most $h\Delta^d$ additional vertices with respect to G''.

We just showed that an instance of MAX H-FREE SUBGRAPH, under deletion of h vertices is equivalent to an instance of the problem under insertion of $h\Delta^d$ vertices, which concludes the proof. ∎

Recall that, for the case of insertion, another generic algorithm was proposed in [3]. This algorithm, denoted by `R2` uses a polynomial ρ-approximation algorithm for the deterministic problem as subroutine to improve the approximation ratio for the reoptimization version from $\frac{1}{2}$ to $\frac{1}{2-\rho}$. However, considering that most hereditary problems are not constant-approximable in polynomial time (unless $\mathbf{P} = \mathbf{NP}$), `R2` cannot be implemented in general graphs.

Note that, u*nder vertex-deletion,* MAX INDEPENDENT SET *in bounded-degree graphs is approximable within ratio 1/2* [12]. Regarding this result, and considering that MAX INDEPENDENT SET is $3/(\Delta + 2)$-approximable in graphs of maximum degree Δ, Algorithm `R2` can be implemented in the vertex-deletion setting. Indeed, the following result, proved in Appendix F, improves the result of [12] just claimed in italics and concludes the paper.

Proposition 11. *In graphs of degree bounded by Δ, under deletion of h vertices,* MAX INDEPENDENT SET *is approximable within ratio $\frac{\Delta+2}{2\Delta+1}$ in polynomial time.*

References

[1] C. Archetti, L. Bertazzi, and M.G. Speranza. Reoptimizing the traveling salesman problem. *Networks*, 42(3):154–159, 2003.

[2] C. Archetti, L. Bertazzi, and M.G. Speranza. Reoptimizing the 0-1 knapsack problem. *Discrete Applied Mathematics*, 158(17):1879–1887, 2010.

[3] G. Ausiello, V. Bonifaci, and B. Escoffier. Complexity and approximation in reoptimization. *CiE 2007: Logic and Computation and Logic in the Real World*, 2007.

[4] G. Ausiello, B. Escoffier, J. Monnot, and V.Th. Paschos. Reoptimization of minimum and maximum traveling salesman's tours. In L. Arge and R. Freivalds, editors, *SWAT*, volume 4059 of *Lecture Notes in Computer Science*, pages 196–207. Springer, 2006.

[5] D. Bilò, H.-J. Böckenhauer, J. Hromkovic, R. Královic, T. Mömke, P. Widmayer, and A. Zych. Reoptimization of steiner trees. In J. Gudmundsson, editor, *SWAT*, volume 5124 of *Lecture Notes in Computer Science*, pages 258–269. Springer, 2008.

[6] D. Bilò, P. Widmayer, and A. Zych. Reoptimization of weighted graph and covering problems. In E. Bampis and M. Skutella, editors, *WAOA*, volume 5426 of *Lecture Notes in Computer Science*, pages 201–213. Springer, 2008.

[7] H.-J. Böckenhauer, L. Forlizzi, J. Hromkovic, J. Kneis, J. Kupke, G. Proietti, and P. Widmayer. On the approximability of tsp on local modifications of optimally solved instances. *Algorithmic Operations Research*, 2(2):83–93, 2007.

[8] H.-J. Böckenhauer, J. Hromkovic, R. Královic, T. Mömke, and P. Rossmanith. Reoptimization of steiner trees: Changing the terminal set. *Theor. Comput. Sci.*, 410(36):3428–3435, 2009.

[9] H.-J. Böckenhauer, J. Hromkovic, T. Mömke, and P. Widmayer. On the hardness of reoptimization. In V. Geffert, J. Karhumäki, A. Bertoni, B. Preneel, P. Návrat, and M. Bieliková, editors, *SOFSEM*, volume 4910 of *Lecture Notes in Computer Science*, pages 50–65. Springer, 2008.

[10] H.-J. Böckenhauer and D. Komm. Reoptimization of the metric deadline tsp. *J. Discrete Algorithms*, 8(1):87–100, 2010.

[11] R. Boppana and M.M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, 32(2):180–196, 1992.

[12] N. Boria, J. Monnot, and V. Th. Paschos. Reoptimization of maximum weight induced hereditary subgraph problems. Cahier du LAMSADE 311, LAMSADE, UniversitÃľ Paris-Dauphine, Juin 2001.

[13] N. Boria and V.Th. Paschos. Fast reoptimization for the minimum spanning tree problem. *Journal of Discrete Algorithms*, 8(3):296–310, 2010.

[14] B. Escoffier, M. Milanic, and V.Th. Paschos. Simple and fast reoptimizations for the steiner tree problem. *Algorithmic Operations Research*, 4(2):86–94, 2009.

[15] U. Feige. Approximating maximum clique by removing subgraphs. *SIAM Journal on Discrete Mathematics*, 18:219, 2004.

[16] M.M. Halldórsson and J. Radhakrishnan. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica*, 18(1):145–163, 1997.

[17] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. *Automata, Languages and Programming*, pages 40–51, 1993.

[18] N. Robertson and P.D. Seymour. Graph minors. xx. wagner's conjecture. *J. Comb. Theory, Ser. B*, 92(2):325–357, 2004.

[19] A. Wagner. On finite affine line transitive planes. *Mathematische Zeitschrift*, 87(1):1–11, 1965.

[20] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proc. STOC'06*, pages 681–690, 2006.

A Proof of Proposition 3

Consider a graph H, instance of MAX INDEPENDENT SET, and denote by α the independence number associated with this instance. We build in polynomial time a graph H_k, instance of MAX k-COLORABLE SUBGRAPH, by duplicating k times the instance of MAX INDEPENDENT SET, and connecting all pairs of vertices from different copies. The graph associated with this instance is denoted $H_k(V, E)$. In H_k, an optimal k-colorable subgraph has weight exactly $k\alpha$. Indeed, given the structure of instance H_k, it holds that its independence number is the same as H (namely α), so that no k-colorable subgraph can weight more than $k\alpha$. On the other hand, taking an optimal independent set in each copy produces a k-colorable subgraph with weight exactly $k\alpha$. We build a reoptimization instance $I_{\alpha,k,h}$ of MAX k-COLORABLE SUBGRAPH as follows:

- The initial graph $G_{\alpha,k,h}$ is obtained by adding to the graph H_k a clique X of k vertices $X = (x_1, \ldots, x_k)$, each with weight α, and connecting all these vertices to all vertices of V. In this graph, an optimal k-colorable subgraph is given by the k-clique X, and this solution will be the initial optimum used in the reoptimization instance.
- The perturbed graph $G'_{\alpha,k,h}$ is obtained by adding a clique Y of h vertices $y_1, \ldots, y_h$ to $G_{\alpha,k,h}$, also with weight α, and connecting all these vertices to all vertices in X.

Denote by OPT' an optimal k-colorable subgraph in $G'_{\alpha,k,h}$. An optimal k-colorable subgraph in Y has weight at most $\min\{h, k\}\alpha$, and, considering that Y is disconnected from V, the union of a k-colorable subgraph in Y and one in V is also k-colorable. Recall that an optimal k-colorable subgraph in V has weight $k\alpha$, then it holds that $w(\text{OPT}') \geqslant (k + \min\{h, k\})\alpha$.

Denote by `A` a $\max\left\{\frac{k}{k+h}, \frac{1}{2}\right\} + \varepsilon$ approximation algorithm for the reoptimization of MAX k-COLORABLE SUBGRAPH under the insertion of h vertices. Let $S_{\alpha,k,h}$ be the solution returned by `A` on the reoptimization instance $I_{\alpha,k,h}$ we just described. It holds that:

$$w(S_{\alpha,k,h}) \geqslant \left(\frac{k}{k + \min\{h, k\}} + \varepsilon\right) \text{OPT}' \geqslant (1 + \varepsilon)k\alpha$$

However, considering that $X \cup Y$ is a clique on $k + h$ vertices, each with weight α , then the restriction of $S_{\alpha,k,h}$ to $X \cup Y$ cannot have weight more than $k\alpha$. Hence, $w(S_{\alpha,k,h}[V]) = w(S_{\alpha,k,h}) - w(S_{\alpha,k,h}[X \cup Y] \geqslant \varepsilon k\alpha$.

Now, notice that $S_{\alpha,k,h}[V]$ is a partitioned in at most k independent sets, the biggest of which has weight at least $\varepsilon\alpha$, and is constrained to be included in a single copy of the original instance of MAX INDEPENDENT SET. Thus, building n reoptimization instances $I_{i,k,h}$ $(1 \leqslant i \leqslant n)$, and applying algorithm `A` on each of them, one can find in polynomial time an independent set of size at least $\varepsilon\alpha$ in the original instance of MAX INDEPENDENT SET, which is impossible unless $\mathbf{P} = \mathbf{NP}$.

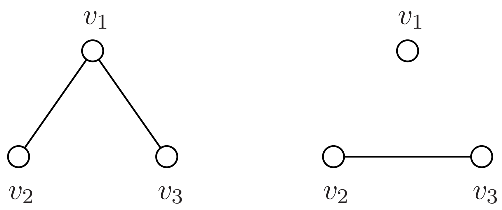
B Proof of Lemma 1

If G is a clique, then $w(G_C) = w(G)$ and $w(G_S) \geqslant \frac{w(G)}{h}$, and symmetrically, if G is an independent set, then $w(G_S) = w(G)$ and $w(G_C) \geqslant \frac{w(G)}{h}$. In both cases, the proposition is verified.

If G is neither a clique nor an independent set (which might occur when $h = 3$), then there are only two possible configurations, both represented in Figure 2.

In Case 1, there are two maximal cliques: $\{v_1, v_2\}$ and $\{v_1, v_3\}$, so $w(G_C) \geqslant \frac{2w(v_1)+w(v_2)+w(v_3)}{2}$. On the other hand there are two maximal independent sets: $\{v_1\}$ and $\{v_2, v_3\}$, so $w(G_S) \geqslant \frac{1}{4}w(v_1) + \frac{3}{4}(w(v_2) + w(v_3))$. In all: $w(G_C) + w(G_S) \geqslant \frac{5w(v_1)+5w(v_2)+5w(v_3)}{4} = \frac{5}{4}w(G)$.

Taking into account the symmetry between Cases 1 and 2, the same bound holds for Case 2, which concludes the proof.



(a) Case 1 (b) Case 2

Figure 2: Graphs G that are neither cliques nor independent sets

C Proof of Proposition 6

C.1 Construction of the initial instance H_{α,β,h_1,h_2}

Consider an unweighted graph H where one wishes to solve MAX SPLIT SUBGRAPH. Denote by α its independence number and by β its clique number. Now, suppose that one builds a graph $H_{\alpha,\beta}$ by adding to H two graphs H_1 and H_2, that are complementary graphs of H, and connecting all vertices in H to vertices in H_2, while H_1 is disconnected from H. Between H_1 and H_2, vertices are connected in the following way:

- if $\alpha < \beta$, then all vertices in H_1 are connected to all vertices in H_2,
- if $\alpha \geqslant \beta$, H_1 and H_2 are completely disconnected.

In $H_{\alpha,\beta}$, both the clique and independence numbers are $\alpha + \beta$. Indeed, Notice that each clique in H becomes an independent set in both H_1 and H_2, and vice versa.

If a clique C takes at least one vertex in H_1, then, either $\alpha \geqslant \beta$ and this clique must be included in H_1 ($|C| \leqslant \alpha$), or $\alpha < \beta$, and this clique can have vertices in H_1 and H_2 but not in H, so that $|C| \leqslant 2\alpha < \alpha + \beta$. On the other hand, if C takes no vertex in H_1, then it must be composed of a clique in H (of size at most β) and a clique in H_2 (of size at most α), so that the clique number of $H_{\alpha,\beta}$ is exactly $\alpha + \beta$.

By symmetry, the same holds for the independence number.

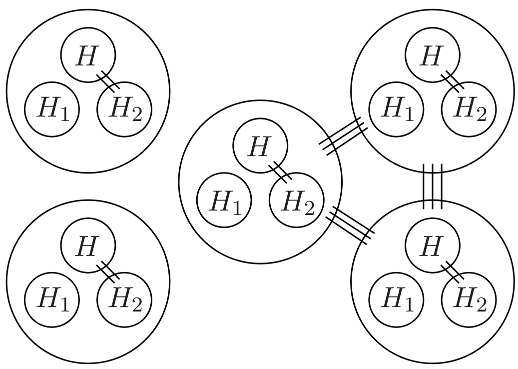


Figure 3: The graph $H_{\alpha,\beta,3,3}$ ($\alpha \geqslant \beta$)

In what follows, we note $\gamma = \alpha + \beta$. Consider the graph H_{α,β,h_1,h_2} ($h_1, h_2 \geqslant 1$) that consists of $h_1 + h_2 - 1$ copies of the graph $H_{\alpha,\beta}$. Among them, h_1 copies are disjoint from one another, and h_2 copies are connected by all possible pairs of vertices from different copies (an example is provided in Figure 3). It holds that:

- H_{α,β,h_1,h_2} has independence number $h_1\gamma$,
- H_{α,β,h_1,h_2} has clique number $h_2\gamma$,
- An ε-approximate solution for MAX SPLIT SUBGRAPH in H_{α,β,h_1,h_2} can be easily derived into a $(2\varepsilon/3)$-approximate solution for MAX SPLIT SUBGRAPH in H. Indeed, any solution SG on H_{α,β,h_1,h_2} can be partitioned into $3(h_1+h_2-1)$ feasible solutions on H, the biggest of which has weight at least $(2\varepsilon/3)\gamma$ when SG is an ε-approximate solution. Thus, MAX SPLIT SUBGRAPH is inapproximable within any constant ratio in H_{α,β,h_1,h_2}, unless $\mathbf{P} = \mathbf{NP}$.

In what follows, we will build three different reoptimization instances, for cases $h \leqslant 2$ (Appendix C.2), $h = 3$ (Appendix C.3) and $h \geqslant 4$ (Appendix C.4). In the three cases, the initial graphs will have the same generic structure, defined as G_{α,β,h_1,h_2} (h_1 and h_2 will take different values in the three specific cases).

This graph is obtained by adding to H_{α,β,h_1,h_2} a set X of vertices, which consists of a clique X_C of size $h_2\gamma + 1$ and an independent set X_S of size $h_1\gamma + 1$. Each vertex in the graph receives weight 1. Clique X_C is disconnected from V, and each vertex of X_S is connected to all other vertices of the graph, namely vertices of both V and X_C. In this initial graph, it holds that a split-graph SG has weight at most $(h_1 + h_2)\gamma + 2$. We distinguish the three following cases.

Case 1. $SG \cap V = \emptyset$, then $w(S) \leqslant w(X) \leqslant (h_1 + h_2)\gamma + 2$.

Case 2. $|SG \cap V| = 1$, and denote by v the single vertex of $SG \cap V$. In this case, SG cannot take more than $\max\{h_1, h_2\}\gamma + 1$ vertices in X. Indeed, if it takes more vertices, then it means that SG contains at least two vertices of each set X_S and X_C, which necessarily forms a forbidden subgraph along with v. It means that $w(SG) \leqslant \max\{h_1, h_2\}\gamma + 2 \leqslant (h_1 + h_2)\gamma + 2$.

Case 3. $|SG \cap V| \geqslant 2$. Denote by SG_C the clique in SG and by SG_S the independent set in SG.

If $SG[V]$ is a clique, then the clique in SG, SG_C, can contain at most $h_2\gamma$ vertices in V and one vertex in X_S, so that $w(SG_C) \leqslant h_2\gamma + 1$. On the other hand, the biggest independent set in X is X_S, so that $w(SG_S) \leqslant w(S_X) \leqslant h_1\gamma + 1$. In all:

$$w(SG) = w(SG_S) + w(SG_C) \leqslant (h_1 + h_2)\gamma + 2$$

A symmetrical arguments holds when $SG[V]$ is an independent set.

Suppose now that $SG[V]$ is neither a clique nor an independent set. Then $w(SG[V]) \leqslant (h_1 + h_2)\gamma$, and both SG_S and SG_C contain at least one vertex in V. Thus, SG_C cannot contain any vertex of X_C, and at most one of vertex of X_S. Symmetrically, SG_S cannot contain any vertex of X_S, and at most one of vertex of X_C. In all, $w(SG[X]) \leqslant 2$, and once more, $w(SG) \leqslant (h_1 + h_2)\gamma + 2$, concluding so Case 3.

Thus, any solution that has weight exactly $(h_1 + h_2)\gamma + 2$ is optimal in G_{α,β,h_1,h_2}, so in all reoptimization instances we will build, we can consider X as the initial optimum.

C.2 The case $h \leqslant 2$

Assume $h \leqslant 2$. We build a reoptimization instance, $I_{\alpha,\beta,h}$ in the following way:

- The initial graph is the graph $G_{\alpha,\beta,h,1}$. We proved in Appendix C.1 that X is an optimum on this graph. Here, its weight is $(h + 1)\gamma + 2$.
- The perturbed graph $G'_{\alpha,\beta,h,1}$ is obtained by adding a set of vertices Y to $G_{\alpha,\beta,h,1}$, which consists of an independent set of h vertices, each with weight γ. All vertices in Y are connected to all vertices in X_S only.

The overall structure is represented in Figure 1, as well as the weight of optimal independent sets and cliques (denoted by S^* and C^* in the Figure) in all sets V, X_C, X_S and Y.

Notice that, in the perturbed graph $Y \cup X_C \cup F^*$ (where F^* is an optimal independent set in V) defines an split graph of weight $(2h+1)\gamma+1$. Indeed, $Y \cup F^*$ defines an independent set, while X_C defines an clique. Thus, denoting by OPT' an optimal split graph in $G'_{\alpha,\beta,h,1}$, it holds that $w(\mathrm{OPT}') \geqslant (2h+1)\gamma+1$.

Suppose that, for a given $h \leqslant 2$, there exists an approximation algorithm `A` for the reoptimization version of MAX SPLIT SUBGRAPH, which provides an approximation ratio bounded by $\frac{h+1}{2h+1}+\varepsilon$, under the insertion of h vertices. Denoting by $S_{\alpha,\beta,h}$ a solution returned by this algorithm on the reoptimization instance $I_{\alpha,\beta,h}$ we just described, it holds that:

$$w(S_{\alpha,\beta,h}) \geqslant \left(\frac{h+1}{2h+1}+\varepsilon\right)\mathrm{OPT}' \geqslant (1+\varepsilon)(h+1)\gamma$$

However, a split graph SG in $X \cup Y$ (and a fortiori the restriction of $S_{\alpha,\beta,h}$ to $X \cup Y$, denoted by $S_{\alpha,\beta,h}[X \cup Y]$) cannot have weight more than $(h+1)\gamma+2$. We distinguish here the following two cases.

Case 1. SG takes at most one vertex in X_S, then $w(SG[X_S]) \leqslant 1$, and thus:

$$w(SG) = w(SG[X_S]) + w(X_C) + w(Y) \leqslant 1+\gamma+1+h\gamma = (h+1)\gamma+2$$

Case 2. SG takes at least two vertices in X_S, then the independent set in SG can contain only vertices of X_S. In other words, the vertices of $Y \cup X_C$ can only be part of the clique in SG. It is quite obvious that the biggest clique in $Y \cup X_C$ is X_C itself so that in this case $w(SG) \leqslant w(X) = (h+1)\gamma+2$. One immediately derives from this result that $w(S_{\alpha,\beta,h}[X \cup Y]) \leqslant (h+1)\gamma+2$ and Case 2 is concluded.

So, in both cases it holds that $w(S_{\alpha,\beta,h}[V]) = w(S_{\alpha,\beta,h}) - w(S_{\alpha,\beta,h}[X \cup Y]) \geqslant \varepsilon\gamma - 2$. Considering that γ is not a constant, if an algorithm `A` exists, one can get in polynomial time a constant-approximate solution for MAX SPLIT SUBGRAPH in the graph $H_{\alpha,\beta,h,1}$, which is impossible unless $\mathbf{P} = \mathbf{NP}$.

C.3 The case $h = 3$

Suppose now that $h = 3$. In this case, the reoptimization instance $I_{\alpha,\beta,h}$ is built as follows:

- The initial graph is the graph $G_{\alpha,\beta,2,3}$. X is an optimum on this graph. Here, its weight is $5\gamma+2$.
- The perturbed graph $G'_{\alpha,\beta,2,3}$ is obtained by adding a path Y to $G_{\alpha,\beta,2,3}$, which has three vertices y_1, y_2 and y_3; y_1 is the central vertex of the path, and has weight 2γ, while y_2 and y_3 are end vertices of the path, and have both weight γ. All vertices in Y are connected to all vertices in X_S, but not to any vertex in X_C. y_1 is connected to all vertices in V, while vertices y_2 and y_3 are not connected to any vertex in V.

Notice that, in the perturbed graph $Y \cup F^*$ (where F^* is an induced optimal split subgraph in V) defines a split graph of weight at least $9\gamma-1$ (in V the maximum clique F^*_C and the maximum independent set F^*_S might have one vertex in common). Indeed, $y_1 \cup F^*_C$ defines a clique, while $y_2 \cup y_3 \cup F^*_S$ defines an independent set. Thus, denoting by OPT' an optimal split graph in $G'_{\alpha,\beta,h}$, it holds that $w(\mathrm{OPT}') \geqslant 9\gamma-1$.

Assume that there exists an approximation algorithm `A` for the reoptimization version of MAX SPLIT SUBGRAPH, which provides an approximation ratio bounded by $\frac{5}{9}+\varepsilon$, under the insertion of $h = 3$ vertices. Denoting by $S_{\alpha,\beta,h}$ a solution returned by this algorithm on the reoptimization instance $I_{\alpha,\beta,h}$ we just described, it holds that:

$$w(S_{\alpha,\beta,h}) \geqslant \left(\frac{5}{9} + \varepsilon\right) \mathrm{OPT}' \geqslant (1 + 9\varepsilon)5\gamma - \frac{5}{9} - \varepsilon \geqslant (1 + \varepsilon)5\gamma \tag{3}$$

where the last inequality follows from noticing that $5/9+\varepsilon < 8\varepsilon\gamma$, otherwise a single vertex defines an 8ε-approximate solution in V, which is supposed to be impossible to provide in polynomial time, unless $\mathbf{P} = \mathbf{NP}$.

However, it holds that a split graph SG in $X \cup Y$ has weight at most $5\gamma + 3$ and we study the following four cases.

Case 1. $SG[Y] = \emptyset$. Then $w(SG) \leqslant w(X) = 5\gamma + 2$,

Case 2. $|SG[Y] = 1|$. If the single vertex is part of the clique in SG, then this clique can take at most one additional vertex in X_S. Symmetrically, if this vertex is part of the independent set in SG, then this independent set can take at most one additional vertex in X_C:

$$w(SG) \leqslant w(y_1) + 1 + \max\{w(X_C), w(X_S)\} \leqslant 5\gamma + 3$$

Case 3. $|SG[Y] = 2|$. If $SG[Y]$ has two connected vertices, then:

- $w(SG[Y]) = 3\gamma$;
- the clique in SG can take at most one vertex in X_S, and none in X_C;
- the independent set in SG cannot have weight more than $w(X_S) = 2\gamma + 1$.

If, on the other hand, $SG[Y]$ has two disconnected vertices, then:

- $w(SG[Y]) = 2\gamma$;
- the independent set in SG can take at most one vertex in X_C, and none in X_S;
- the clique in SG cannot have weight more than $w(X_C) = 3\gamma + 1$.

Hence, in both cases, one verifies that $w(SG) \leqslant 5\gamma + 3$.

Case 4. $|SG[Y] = 3|$. In this case, $w(SG[Y]) = 4\gamma$, and SG can take at most two vertices of X, one of X_S that can be part of the clique in SG, of X_C that can be part of the independent set in SG. In all, $w(SG) \leqslant 4\gamma + 2$, that concludes Case 4.

Taking into account that any split graph in $X \cup Y$ cannot have weight more than $5\gamma + 3$, the same holds a fortiori for $S_{\alpha,\beta,h}[X \cup Y]$. Combining this bound with (3), one immediately derives that:

$$w(S_{\alpha,\beta,h}[V]) \geqslant (1 + \varepsilon)5\gamma - 5\gamma - 3 = \varepsilon\gamma - 3 \tag{4}$$

For reasons already explained, (4) makes impossible the existence of a polynomial algorithm `A` that ensures a $\frac{5}{9} + \varepsilon$ approximation ratio under the insertion of $h = 3$ vertices, unless $\mathbf{P} = \mathbf{NP}$.

C.4 The case $h \geqslant 4$

Finally, suppose that $h \geqslant 4$. We build the following reoptimization instance $I_{\alpha,\beta,h}$:

- The initial graph is the graph $G_{\alpha,\beta,1,1}$. In it, X defines an optimal split graph of weight $2\gamma + 2$.

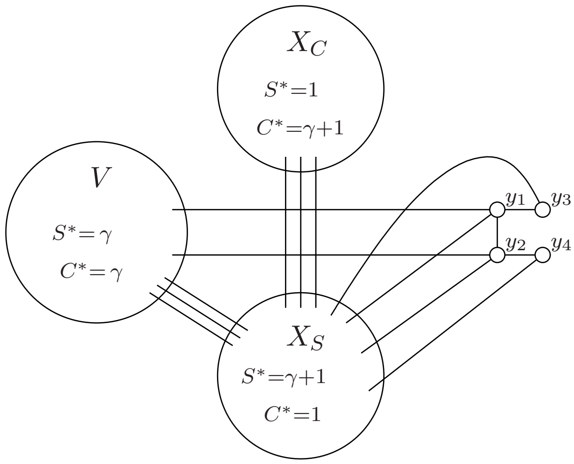


Figure 4: The reoptimization instance $I_{\alpha,\beta,h}$, $h \geqslant 4$

- The perturbed graph $G'_{\alpha,\beta,h}(V_{\alpha,\beta,h}, E_{\alpha,\beta,h})$ is obtained by adding a set Y of h vertices, in which 4 vertices have weight $\gamma/2$, and are as represented in Figure 4: all these vertices are connected to all vertices in X_S, while only y_1 and y_2 are connected to all vertices in V. The other $h-4$ vertices in Y have null weight, and will be ignored in what follows.

Notice that in V, an optimal split graph F^* has weight at least $2\gamma - 1$ (since a clique and independent set can have at most one vertex in common), notice also that $Y \cup F^*$ defines a feasible solution: the clique in F^* forms a clique along with y_1 and y_2, and the independent set in F^* forms an independent set along with y_3 and y_4. Thus, denoting by OPT' an optimal solution in the perturbed graph $G'_{\alpha,\beta,h}$, it holds that $w(\text{OPT}') \geqslant 4\gamma - 1$.

Suppose that, for a given $h \geqslant 4$, there exists an approximation algorithm `A` for the reoptimization version of MAX SPLIT SUBGRAPH, which provides an approximation ratio bounded by $\frac{1}{2} + \varepsilon$, under the insertion of h vertices. Denoting by $S_{\alpha,\beta,h}$ a solution returned by this algorithm on the reoptimization instance $I_{\alpha,\beta,h}$ we just described, its weight can be bounded as follows:

$$w(S_{\alpha,\beta,h}) \geqslant \left(\frac{1}{2} + \varepsilon\right) w(\text{OPT}') \geqslant (1+2\varepsilon)2\gamma - \frac{1}{2} - \varepsilon \geqslant (1+\varepsilon)2\gamma \tag{5}$$

where the last inequality follows from noticing that $1/2 + \varepsilon < 2\varepsilon\gamma$. If not, then a single vertex defines a ε-approximate solution in V.

We prove now that a split graph SG in $X \cup Y$ (and a fortiori the restriction of $S_{\alpha,\beta,h}$ to this set) cannot have weight bigger than $2\gamma + 2$. For this, we study the following three cases.

Case 1. $|SG \cap Y| = 0$. Obviously $w(SG) \leqslant w(X) = 2\gamma + 2$,

Case 2. $|SG \cap Y| = 1, 2$. SG can take at most $\gamma + 2$ vertices in X, otherwise, it takes at least 2 vertices in both X_C and X_S, which form a forbidden subset along with one vertex in Y. Thus, $w(SG) \leqslant \gamma + 4 \leqslant 2\gamma + 2$

Case 3. $|SG \cap Y| \geqslant 3, 4$. Given the structure of Y, then 3 (and a fortiori 4) vertices cannot form an independent set, nor a clique. Thus, SG can take at most one vertex in each set X_C and X_S, and $w(SG) \leqslant 2\gamma + 2$ and Case 3 is concluded.

Hence, the restriction of $S_{\alpha,\beta,h}$ to $X \cup Y$ has weight bounded as follows:

$$w(S_{\alpha,\beta,h}[X \cup Y]) \leqslant \frac{2}{\gamma} + 2 \tag{6}$$

Combining (5) and (6), one easily derives:

$$w(S_{\alpha,\beta,h}[V]) \geqslant (1+\varepsilon)2\gamma - 2\gamma - 2 = \varepsilon\gamma - 2 \tag{7}$$

So, $S_{\alpha,\beta,h}[V]$ is a constant-approximate solution for MAX SPLIT SUBGRAPH in the graph $H_{\alpha,\beta,1,1}$, which is impossible to provide in polynomial time unless $\mathbf{P} = \mathbf{NP}$.

D Proof of Proposition 8

Consider a reoptimization instance I of the MAX k-COLORABLE SUBGRAPH problem, under deletion of $h < k$ vertices. The initial graph is denoted by $G(V, E)$, and the perturbed one by $G'(V', E')$ where $G' = G[V \setminus Y]$. Let OPT and OPT$'$ denote optimal k-colorable subgraphs on G and G' respectively. Considering that Y has h vertices, and denoting by h' the number of independent sets in OPT that contain at least one vertex of Y, it holds that $h' \leqslant h$.

Let S_1 denote the $(k - h')$-colorable subgraph in OPT which does not contain any vertex of Y and S_2 the h'-colorable subgraph in OPT such that each independent set in it has at least one vertex in Y. Consider the simple reoptimization algorithm that consists of returning the set SOL $=$ OPT $\setminus Y$ (that is, the remaining part of the optimum after the deletion of Y). It holds that:

$$\begin{aligned} w(\mathrm{SOL}) &= w(S_1) + w(S_2 \setminus Y) && (8)\\ w(\mathrm{OPT}') &\leqslant w(\mathrm{OPT}' \setminus S_2) + w(S_2 \setminus Y) && (9) \end{aligned}$$

It also holds that S_1 is an optimal $(k - h')$-colorable subgraph in the induced subgraph $G[V \setminus S_2]$ (otherwise OPT wouldn't be an optimal solution). It also holds that OPT$' \setminus S_2$ defines a k-colorable subgraph in $G[V \setminus S_2]$. Thus, the $k - h'$ biggest independent sets in OPT$' \setminus S_2$ have weight at most $w(S_1)$, and at least $\frac{k-h}{k} w(\mathrm{OPT}' \setminus S_2)$. Hence, one verifies that:

$$w(S_1) \geqslant \frac{k - h'}{k} w(\mathrm{OPT}' \setminus S_2) \tag{10}$$

Combining (8), (9) and (10), one finally proves that:

$$\frac{\mathrm{SOL}}{\mathrm{OPT}'} \geqslant \frac{\frac{k-h'}{k} w(\mathrm{OPT}' \setminus S_2) + w(S_2 \setminus Y)}{w(\mathrm{OPT}' \setminus S_2) + w(S_2 \setminus Y)} \geqslant \frac{k-h}{k}$$

that concludes the proof.

E Proof of Proposition 9

Revisit the proof of Proposition 3 in Appendix A: out of an instance H of MAX INDEPENDENT SET, with independence number α, we can build an instance $H_\alpha(V, E)$ of MAX k-COLORABLE SUBGRAPH, such that for any $i \leqslant k$, an optimal i-colorable subgraph in H_α has weight exactly $i\alpha$, and any constant approximation for MAX k-COLORABLE SUBGRAPH is impossible in H_α, unless $\mathbf{P} = \mathbf{NP}$. We build the following reoptimization instance $I_{\alpha,h}$ $(h < k)$ of MAX k-COLORABLE SUBGRAPH:

- The initial graph $G_{\alpha,h}$ is obtained by adding to H_α a clique Y of k vertices, each connected to all vertices of H_α, and each with weight α.
- The perturbed graph $G'_{\alpha,h}$ is obtained by deleting h of the k vertices of Y. Denote by Y' the set of remaining vertices of Y after the deletion ($|Y'| = k - h$).

For reasons already explained in the proof of Proposition 7, it holds that Y can be considered as the initial optimum of the reoptimization instance.

Suppose that, for a given h, there exists an algorithm $\mathtt{A}$ that computes a $\frac{k-h}{k}+\varepsilon$ approximation for MAX k-COLORABLE SUBGRAPH under deletion of h vertices. And let $S_{\alpha,h}$ denote the solution returned by $\mathtt{A}$ on the instance $I_{\alpha,h}$, it holds that $w(S_{\alpha,h}) \geqslant (1+\varepsilon)(k-h)\alpha$, and considering that $w(Y') = (k-h)\alpha$, it holds that $w(S_{\alpha,h}[V]) \geqslant \varepsilon(k-h)\alpha$.

Thus, an algorithm $\mathtt{A}$ cannot exist for any h, otherwise it would provide a constant approximation for MAX k-COLORABLE SUBGRAPH in the graph $H_\alpha(V,E)$, which is impossible to provide in polynomial time unless $\mathbf{P} = \mathbf{NP}$.

F Proof of Proposition 11

Let $G(V,E)$ be an instance of MAX INDEPENDENT SET, with one known optimal solution OPT, and let $G'(V',E') = G[V \setminus Y]$, $(|Y| = h)$. On this perturbed instance, an optimal solution is denoted by OPT$'$. In what follows, $N(S)$ is the set of neighbors of all the vertices in S. Denote by $\mathtt{A}_\rho$ a ρ-approximation algorithm for MAX INDEPENDENT SET in graphs of bounded degree, and $\mathtt{A}_\rho(G)$ a solution returned by $\mathtt{A}_\rho$ on a given graph G. Consider now the following algorithm:

- let $\mathrm{SOL}_1 = \mathrm{OPT} \setminus Y$;
- for each maximal independent set S_i in $N(\mathrm{OPT} \cap Y)$ (they can all be enumerated in $O(2^{h\Delta})$), set $\mathrm{SOL}_{2,i} = S_i \cup \mathtt{A}_\rho(G'[V' \setminus (S_i \cup N(S_i))])$, and $\mathrm{SOL}_2 = \max_i(\mathrm{SOL}_{2,i})$.

It holds that:

$$w(\mathrm{SOL}_1) \geqslant w(\mathrm{OPT}'[V' \setminus N(\mathrm{OPT} \cap Y)]) \geqslant w(\mathrm{OPT}') - w(\mathrm{OPT}' \cap N(\mathrm{OPT} \cap Y)) \tag{11}$$

On the other hand, notice that OPT$'$ can be divided in two parts, $\mathrm{OPT}' \cap N(\mathrm{OPT} \cap Y)$, and $\mathrm{OPT}' \setminus N(\mathrm{OPT} \cap Y)$. Among all sets S_i's computed in the second part of the algorithm, one must have computed the set $S_{i^*} = \mathrm{OPT}' \cap N(\mathrm{OPT} \cap Y)$. Moreover, the second part of the solution, $\mathtt{A}\rho(G'[V' \setminus (S_{i^*} \cup N(S_{i^*})])$ is a ρ approximation on a subgraph where the optimum is $\mathrm{OPT}' \setminus N(\mathrm{OPT} \cap Y)$. In all:

$$\begin{aligned} w(\mathrm{SOL}_2) \geqslant \mathrm{SOL}_{2,i^*} &\geqslant w(\mathrm{OPT}' \cap N(\mathrm{OPT} \cap Y)) + \rho w(\mathrm{OPT}' \setminus N(\mathrm{OPT} \cap Y) \\ &\geqslant \rho w(\mathrm{OPT}') + (1-\rho) w(\mathrm{OPT}' \cap N(\mathrm{OPT} \cap Y)) \end{aligned} \tag{12}$$

Combining (11) and (12), with coefficients $(1-\rho)$ and 1, one finally proves that:

$$\frac{w(\mathrm{SOL})}{w(\mathrm{OPT}')} \geqslant \frac{1}{2-\rho} \tag{13}$$

Replacing ρ with $3/(\Delta+2)$ in (13) leads to the claimed result.