# CAHIER DU LAMSADE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
(Université Paris-Dauphine)
Unité de Recherche Associée au CNRS n° 825

## RELATIVE HARDNESS OF CONSTRUCTIVE–NON-CONSTRUCTIVE APPROXIMATION: THE CASE OF MAXIMUM INDEPENDENT SET PROBLEM

CAHIER N° 137
avril 1996

V.Th. PASCHOS [1]
M. DEMANGE [1]

received: October 1995.

[1] *LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France.*

## CONTENTS

# Difficulté relative de l'approximation constructive et non-constructive: le cas du problème de stable maximum d'un graphe

**Résumé.** Nous proposons une démarche générale pour étudier la difficulté relative de la détermination d'une solution (exacte ou approchée) d'un problème d'optimisation par rapport à la difficulté du calcul (exact ou approché) de la valeur optimale en introduisant des problèmes incluant, dans leur instance, une information sur la valeur. Dans le cas particulier du problème de stable maximum, cette démarche nous conduit à définir une classe de problèmes du stable dont l'étude de l'approximabilité s'avère particulièrement intéressante.

**Mots-clés:** complexité, approximation polynomiale, stable.

# Relative hardness of constructive – non-constructive approximation: the case of maximum independent set problem

**Abstract.** We show how polynomial approximation theory, developed as the main tool for efficiently solving NP-complete problems, can motivate a great number of questions on the meaning of "problem solution". We propose a general thought process for the study of the relative hardness between determining solutions of combinatorial optimization problems and computating (approximately or exactly) their optimal values, whenever we deal with problems including an information on the optimal value in their instance. In the particular case of the maximum independent set, this thought process leads us to define classes of independent set problems the approximability of which is particularly interesting.

**Keywords:** complexity, polynomial approximation, independent set.

## 1 Constructive – non constructive

In theoretical computer science, problems (a fortiori the NP-complete ones) are originally defined as decisions about the existence of solutions verifying some properties. So in the *decision framework* (denoted by **D** in what follows) a problem is a question whose objective is to answer by *yes* or *no*.

However, the polynomial approximation theory needs a different framework, which we call *optimization framework*, defined below, simultaneously establishing and using the notion of the *feasible solution* and the one of the *optimal value*; both notions are crucial for defining the concept of the *approximate solution* and for characterizing its quality.
So in the optimization framework (denoted by **O**) the instance of a problem $\Pi$ is expressed in terms of an optimization program of the form

$$\begin{cases} \text{opt } f(\mathbf{x}) \\ \quad \mathbf{x} \in \mathcal{C} \end{cases}$$

where opt $\in \{\min, \max\}$, $f$ is the objective function and $\mathcal{C}$ the set of feasible solutions. Then, $\Pi$ consists of **determining an optimal solution $\mathbf{x}^*$** and an algorithm determining it is called *constructive*. An approximation algorithm dealing with this framework (i.e., **determining a feasible solution and guaranteeing a nearness of its value to the value of the optimal one**) will be called *constructive approximation algorithm*. The approximation point of view of **O** will be denoted by **O-ap**.

There exists also another framework, the one of the *optimal value* (denoted by **V**), where the instances are the same as in **O** but the goal is now to **determine the optimal value** and an algorithm doing this will be called *non-constructive*. Here also, one can define a concept of approximation (**V-ap**) and an algorithm determining a feasible **value** and guaranteeing a nearness of this value to the optimal one is called *non-constructive approximation algorithm*.

Up to very recently, research in polynomial approximation theory was mainly focused on the framework **O**. But its development makes that other types of problems (of equally mathematical and epistemological natures) start to be posed in the context of this theory. Among these problems, the problem of the relative hardness between the constructive and non-constructive frameworks, i.e., the links between the approaches **D**, **O** and **V**, starts to interest the resarchers. This question is, up to now, marginally treated and only from an optimality point of view without taking into account approximation considerations, i.e., the links between **O-ap** and **V-ap**. This is what we are doing in this paper; we consider the relative hardness between the constructive and the non-constructive approximations.

One first remark is that a constructive algorithm is in particular non-constructive and, also, that a non-constructice algorithm answers also to the decision problem; consequently, **O** is, at least, as hard as **V**, the latter being, at least as hard as **D**. Similarly, a constructive approximation algorithm is also a non-constructive approximation one; the latter is, in general, not able to answer

to the decision problem; so, **O-ap** is, at least, as hard as **V-ap**. For the cases where a non-constructive algorithm provides an answer for the decision version of a problem $\Pi$, the problem of approximating $\Pi$ with a certain approximation guaranty is NP-complete. Consequently, constructive approximation is, at least, as difficult as the non-constructive one and a natural question one can pose is: *in which sense is constructive optimization (resp., approximation) harder than the non-constructine one?*

In order to apprehend this type of question, either from an exact optimization, or from an approximate optimization points of view, we propose in what follows a general thought process consisting in integrating in the instances of the problems an information carrying over their optimal value, in such a way that the new (modified) problem is a priori **V**-easier[1] than the original one. Let us note that in this case, the recognition of the instances of the derived problems is not necessarily polynomial[2]. So, via this transformation, work in framework **O** allows to better understand the links between constructiveness and non-constructiveness and the question posed just above can now be transformed into the following one: *in what an information about the optimal value of a problem helps in deteminining either the optimal or a good approximate solution for it?*

In this paper we try, in a first time, to draw a (first) formal answer to the questions just posed; then, we investigate the case of one of the most famous NP-complete problems, the maximum independent set.

Let $G = (V, E)$ be a graph of order $n$; an *independent set* is a subset of $V' \subseteq V$ such that whenever $\{v_i, v_j\} \subseteq V'$, $v_i v_j \notin E$, and the *maximum independent set problem (IS)* is to find an independent set of maximum size. A natural generalization of IS is the one where positive integer weights are associated with the vertices of $G$; then, the objective becomes to maximize the sum of weights of the vertices of an independent set; we denote this problem by WIS; we suppose also that the (integer) weights are universal constants (independent of $n$).

*Notations.* In what follows, we shall speak about **D-** (resp., **O-**, **V-**) NP-completeness[3] (resp., approximation) to denote the NP-completeness (resp., approximation) of a given problem in framework **D** (resp. **O**, **V**). Furthermore, we shall use the term **D-** (resp., **O-**, **V-**) easy (resp., hard) to denote the facility (resp., hardness) of a given problem in framework **D** (resp. **O**, **V**).
We consider a connected graph $G = (V, E)$ of order $n$. We denote by $\alpha(G)$ the stability number (cardinality of a maximum independent set) of $G$ and by $w_i$ the weight of the vertex $v_i$, $i = 1, \ldots, n$; as usual, $\Gamma(v)$, $v \in V$, denotes the neighbourhood of $v$. For a set $V' \subseteq V$, we denote by $G[V']$ the subgraph of $G$ induced by $V'$; for a set $A$ of edges, we denote by $T[A]$ the set of vertices, endpoints of the edges of $A$.
Moreover, given an instance $I$ of a problem $\Pi$, we denote by $v(I)$ its optimal

[1] Easier when dealing with **V**.
[2] This type of problems are, in particular, not NPO in the sense of [2].
[3] For a formal definition of the notions of **V-** and **D-**NP-completeness, see [5].

**Algorithm ORSTABLE**
**begin**
$S \leftarrow \emptyset$;
**while** $V \neq \emptyset$ **do**
choose a vertex $v \in V$;
$G_c \leftarrow G[V \setminus \{v\}]$
**if** $\mathbf{VAL}(G) \neq \mathbf{VAL}(G_c)$ **then**
$S \leftarrow S \cup \{v\}$;
$V \leftarrow V \setminus (\{v\} \cup \Gamma(v))$;
$G \leftarrow G[V]$
**else** $G \leftarrow G_c$
**fi**
**od**
**end.**

value. Finally, whenever vector **1** is indexed by a set of vertices, it denotes the characteristic vector of this set. Especially for the four linear programs of section 4.2, since the dimensions of the vectors **1** and **0** are not always the same ones, these vectors will be indexed by their dimension.

## 2 First point of view: information over all instances

*Given a combinatorial optimization problem $\Pi$ and an oracle which, for each instance $I$ of $\Pi$, computes the optimal value of $I$, does there exist a constructive polynomial algorithm $OR\Pi$ for $\Pi$?*[4]

The answer to this question is positive in the case where the decision-version of $\Pi$ is NP-complete ([2]). In other words, in the case of the **D**-NP-completeness of $\Pi$, the frameworks **V** and **O** are, globally, of equivalent hardness.

For example, in the case of IS it is very easy to bring to the fore an algorithm **ORSTABLE** which using an oracle **VAL** computing for a graph $G$, its stability number $\mathbf{VAL}(G)$, returns, in time linear to the order of $G$, a maximum independent set[5] $S$ for $G$. As we can see from algorithm **ORSTABLE**, the existence of **VAL** for every instance of IS allows to transform a global information (stability number) into a local one (form of a maximum independent set) in the particular instance.

For IS, a very interesting open question would be: *if one has recourse to an oracle* **APR** *providing, for a graph $G$, the cardinality $\alpha'(G)$ of an independent*

[4] Let us remark that the existence of a polynomial non-constructive algorithm would allow, by substituting this algorithm to the oracle, to establish a constructive polynomial algorithm.

[5] Here, we exceptionally use the notation $\mathbf{VAL}(G)$ instead for $\alpha(G)$ to indicate the computation of the latter quantity by **VAL**.

*set verifying* $\alpha'(G) \geq \rho\alpha(G)$, *with* $0 < \rho < 1$, *would one be able to construct an independent set with cardinality a (constant) fraction of* $\alpha(G)$*?*
This question (posed in the framework **O-ap**) seems to be harder to answer than the analogous one posed in **O** (where given $\alpha(G)$, the computation of the optimal solution is requested).

The following result constitutes a first partial answer to the above question since it shows how we can use **APR** to determine a non-constructive polynomial time approximation schema.

**Proposition 1.** *For every* $\epsilon \in ]0,1[$, *there exists a polynomial time algorithm which, using* **APR**, *provides, for every graph* $G$, *the cardinality* $\alpha'_\epsilon(G)$ *of an independent set verifying* $\alpha'_\epsilon(G) \geq (1-\epsilon)\alpha(G)$.

*Proof.* Let us fix $\epsilon > 0$ and determine the least integer $m$ such that $\rho^{1/m} \geq 1-\epsilon$. Let us then consider the graph $G^m$ inductively defined as follows:

$$G^1 = G$$
$$G^i = G \times G^{i-1}$$

where $G_1 \times G_2$ denotes the composition of the graphs $G_1$ and $G_2$ ([3]). It can be easily proved (see [3,8]) that every independent set of cardinality $\tilde{\alpha}(G)$ in $G$ can be polynomially transformed into an independent set of cardinality $\tilde{\alpha}(G^m) = \tilde{\alpha}(G)^m$ in $G^m$ and vice-versa. Consequently, $\alpha(G^m) = \alpha(G)^m$. So, by constructing $G^m$ and by using **APR** on it, one obtains the cardinality $\alpha'(G^m)$ of an independent set of $G^m$ verifying $\alpha'(G^m) \geq \rho\alpha(G^m)$. In this case, $\alpha'(G) = \alpha'_\epsilon(G) \geq \rho^{1/m}\alpha(G) \geq (1-\epsilon)\alpha(G)$, q.e.d.

## 3 Second point of view: information linked to a fixed instance

In the previous section, the external information attached to the problem was dynamic and implemented via an oracle supposed to be able to answer for every graph (let us also recall that algorithm **ORSTABLE** requires $O(n)$ distinct calls to **VAL**). Such an information is sufficiently rich, at least when we are interested in optimally solving a problem, to allow the construction of an optimal solution.

Let us now consider a static information on problem's value, information much more restrictive and attached to a fixed instance; in this case, the strongest possible information on the optimal value is, of course, the knowledge of this value[6]. Then, for an optimization problem $\Pi$, we define problem $\tilde{\Pi}$ an instance $\tilde{I}$ of which is a pair $(I, \beta(I))$, where $I$ is an instance of $\Pi$ and $\beta(I)$ its optimal value. Of course, $\tilde{\Pi}$ is **V**-polynomial, since it suffices to read the answer in $\tilde{I}$. In **O**, one has to determine, starting from $(I, \beta(I))$, an optimal solution for $I$. Let us remark here that $I$ and $\tilde{I}$ have the same set of feasible (resp., optimal) solutions and, consequently, deciding on the feasibility of a solution for $\tilde{I}$ is

[6] Naturally, from this point of view, **V-ap** can be discarded.

polynomial whenever this decision is polynomial for $I$. *The study of the relative hardness between* **V** *and* **O** *(resp.,* **O***-ap) for* $\Pi$ *is equivalent to the study of the* **O***- (resp.,* **O***-ap) hardness for* $\tilde{\Pi}$.

We then have the following result.

**Theorem 2.** *Let* $\Pi$ *be an optimization problem and consider an instance* $I$ *of* $\Pi$. *Suppose that there exists a bound* $M(I)$, *polynomial in* $\Pi$*'s size, verifying, for every* $I$, $|\beta(I)| \leq M(I)$. *Then, if* $\tilde{\Pi}$ *admits a constructive exact (resp.,* $\rho$*-approximate, for a fixed constant* $\rho$*) polynomial algorithm* $\mathcal{A}$ *of complexity* $I \mapsto f(I)$ *computable in polynomial time, so does* $\Pi$.

*Proof.* Let us suppose that such an algorithm $\mathcal{A}$ exists. It suffices to apply $\mathcal{A}$, for $f(I)$ elementary steps on the data $(I, k)$, $k \in [-M(I), M(I)]$. This, for each $k$, leads either to a decision that it is impossible to apply $\mathcal{A}$, or to a non-feasible solution (in this case, $k \neq \beta(I)$), or to a feasible one (in particular, when $k = \beta(I)$). Then, we simply have to choose the best objective value's solution among the feasible ones.

Let us note that the constraint on the existence of the bound $M$ is satisfied by the most of the known problems; for IS, for example, one can simply consider $M(I) = n$.

## 4 Classes of problems defined with respect to the optimal value of their instances

The result of section 3 shows that in the case where a static information is given, this is not strong enough to deduce the form of an exact or approximate solution for a problem. In any case, in the two previous sections we have followed thought processes quite "radical", in the sense that we have supposed that the additional information were a posteriori attached to the instances of the problems. However, we can consider another type of static information addressed, this time, to (restricted) classes of instances of a given problem, classes defined using this information. This intermediate thought process can be rich enough, since it allows us to apprehend the boundaries between constructiveness and non-constructiveness.

### 4.1 The König – Egervary graphs

The class of König-Egervary graphs (KE-graphs) is composed of graphs $G$ for which the cardinality of a maximum independent set is equal to the cardinality of a minimum edge covering of $G$. So, we really have here a class of IS instances defined starting from an a priori information concerning the optimal value. In this particular case, the knowledge of this information allows to polynomially **V**-solve IS, since we only have to solve minimum edge covering, and this can be done in polynomial time ([3]). So, IS in KE-graphs is polynomial for the non-constructive framework. Moreover, given that Deming ([7]) has proved that the

recognition of a KE-graph can be done in polynomial time and, furthermore, IS is **O**-polynomial in KE-graphs, the constructive version of the problem is also polynomial.

More recently, Bourjolly et al. ([4]) have extended the class KE to the one **b**-KE, where positive integer weights were considered on the vertices of $G$, and proved that WIS remains polynomially **O**-solvable for this class.

Here, we propose another generalization of the KE-graphs in both weighted and non-weighted cases. For the non-weighted case generalizations are issued from a combinatorial interpretation of the relation maximum independent set – minimum edge covering. For the weighted one, this generalization contains a less restrictive information than for the non-weighted case and is based upon linear programming arguments. As we show, in both weighted and non-weighted cases these generalization preserve the polynomiality of WIS.

### 4.2 Some generalizations of König – Egervary graphs

**The non-weighted case.** A general instance of IS defined by a graph $G = (V, E)$ can be written as a 0-1 linear problem as follows:

$$\mathrm{IS}(G) = \begin{cases} \max \mathbf{1}_n \cdot \mathbf{x} \\ \quad A \cdot \mathbf{x} \leq \mathbf{1}_{|E|} \\ \quad \mathbf{x} \in \{0,1\}^n \end{cases}$$

where $A$ is the edge-vertex incidence matrix of $G$.
Let us denote by $\mathrm{IS}_r$ the following relaxed version of IS:

$$\mathrm{IS}_r(G) = \begin{cases} \max \mathbf{1}_n \cdot \mathbf{x} \\ \quad A \cdot \mathbf{x} \leq \mathbf{1}_{|E|} \\ \quad \mathbf{x} \geq \mathbf{0}_n \end{cases}$$

The dual of $\mathrm{IS}_r$ denoted by $\mathrm{EC}_r$ is

$$\mathrm{EC}_r(G) = \begin{cases} \min \mathbf{1}_{|E|} \cdot \mathbf{x} \\ \quad A^T \cdot \mathbf{x} \geq \mathbf{1}_n \\ \quad \mathbf{x} \geq \mathbf{0}_{|E|} \end{cases}$$

where this problem is denoted by $\mathrm{EC}_r$ in order to indicate that it is the relaxed version of the following problem EC known as the minimum edge covering:

$$\mathrm{EC}(G) = \begin{cases} \min \mathbf{1}_{|E|} \cdot \mathbf{x} \\ \quad A^T \cdot \mathbf{x} \geq \mathbf{1}_n \\ \quad \mathbf{x} \in \{0,1\}^{|E|} \end{cases}$$

Remark that $\mathbf{0}_n$ and $\mathbf{1}_{|E|}$ are feasible for $\mathrm{IS}_r$ and $\mathrm{EC}_r$, respectively. As these dual instances have their respective constraint sets non empty, they have the same optimal value. Then, the following inequalities hold[7]: $\alpha(G) \leq v(\mathrm{IS}_r(G)) =$

[7] Recall that, following our notations, $v(\mathrm{IS}(G)) = \alpha(G)$.

$v(\mathrm{EC}_r(G)) \leq v(\mathrm{EC}(G))$. Let us refer to the difference $v(\mathrm{EC}(G)) - \alpha(G)$ as the *discrete duality gap.*

A very interesting idea seems to be the generalization of the class KE (we will see later how this generalization applies also to the class **b**-KE) by allowing a certain "freedom" to the discrete duality gap. To do that, we first propose a combinatorial interpretation of the difference $v(\mathrm{EC}(G)) - \alpha(G)$.

Given a graph $G$, let us consider a minimum vertex cover $C^*$ ($|C^*| = \tau(G)$) and the corresponding maximum independent set $S^*$. Let us also suppose that, given a maximum matching $M$ ($|M| = m$), there are $f$ matching edges such that both their endpoints belong to $C^*$, for the remaining ones, one of their endpoints belonging to $C^*$ and the other one to $S^*$. Let us call this edges "dissymmetric" and denote by $F$ the set of these "dissymmetric" edges ($|F| = f$). For $M$ (whenever it is not perfect), let us denote by $X$ the set of the exposed (non-saturated) vertices of $G$ with respect to $M$, and by $X_C$ ($|X_C| = g$) and $X_S$ the subsets of $X$ belonging to $C^*$ and $S^*$, respectively (of course, $X = X_C \cup X_S$). The numbers $f$ and $g$, consequently the sets $F$ and $X_C$, depend not only on $M$ but also, for a fixed matching, on the sets $C^*$ (and $S^*$) considered. However, the sum $f + g$ is a quantity depending only on $G$ ($f + g$ is the discrete duality gap). In fact, for every graph $G$, $\tau(G) = m + (f + g)$ and $|S^*| = \alpha(G) = n - m - (f + g)$. Consequently, $v(\mathrm{EC}(G)) - v(\mathrm{IS}(G)) = f + g$ and *a KE-graph is exactly a graph where* $f + g = 0$.

If we relax the constraint $f + g = 0$ by allowing a positive discrete duality gap bounded above, we get the following proposition.

**Proposition 3.** *Consider a graph* $G = (V, E)$ *such that* $0 \leq \tau(G) - m = f + g \leq \kappa$. *Then, (i) if* $\kappa$ *is a fixed positive integer constant, there exists an exact polynomial algorithm for maximum independent set problem in* $G$; *(ii) otherwise, there exists a polynomial time approximation algorithm (having* $\kappa$ *among its input parameters) providing an independent set of cardinality, at least equal to* $\lceil n/[2(\kappa + 1)] - 2 \rceil$.

*Proof.* (i) The condition $f + g \leq \kappa$ implies that both $f$ and $g$ are bounded above by $\kappa$. So, for all integer $h \leq \min\{m, \kappa\}$ and for all integer $k \leq \min\{n - 2m, \kappa - h\}\}$, for all $h$-tuple $H$ of matching edges and for all $k$-tuple $K$ of exposed vertices of $V$ with respect to $M$, we form the sub-graphs of $G$ induced by the vertex set $V \setminus (K \cup T[H])$. We next apply the algorithm of [7] on all of the so-obtained sub-graphs of $G$. Then, for one of the pairs $(H, K)$, the induced subgraph is KE and, moreover, in this graph, a maximum independent set is identical to the maximum independent set of $G$ (by the definition of the quantities $f$ and $g$). Hence, the maximum cardinality set between the so-obtained independent sets is a maximum independent set of $G$. Given that $\kappa$ is a universal constant, there is a polynomial number of pairs $(H, K)$ and, consequently, the whole of the described process remains also polynomial.

(ii) Obviously, $f \leq \kappa$ and $g \leq \kappa$. We arbitrarily partition the edges of $M$ into $\kappa + 1$ sets $M_1, \ldots, M_{\kappa+1}$, where $|M_i| = \lfloor m/(\kappa + 1) \rfloor$, $i = 1, \ldots, \kappa$ and $M_{\kappa+1} = m - \sum_{i=1}^{\kappa} \lfloor m/(\kappa + 1) \rfloor$. We also arbitrarily partition the set $X$ of the

exposed vertices of $G$ into $\kappa+1$ sets $X_1,\ldots,X_{\kappa+1}$ sets, each of size at least $\lfloor (n-2m)/(\kappa+1)\rfloor$. We so obtain $(\kappa+1)^2$ sub-graphs of $G$, each one induced by the vertex-set $X_i \cup T[M_j]$, $(i,j) \in \{1,\ldots,\kappa+1\}^2$ (Cartesian square).
So, we can apply the algorithm of [7] on all of the so-obtained $(\kappa+1)^2$ graphs. Since at least one of these graphs is KE, one of the obtained solutions will be of size at least $(n-2m)/(\kappa+1)+m/(\kappa+1)-2=(n-m)/(\kappa+1)-2$. The minimum of this quantity, for $m \in [0,\frac{n}{2}]$, is obtained for $m=n/2$ and its value is, in this case, equal to $[n/[2(\kappa+1)]]-2$.

**Corollary 4.** *Given a fixed positive constant $\kappa$, deciding if a graph $G$ verifies $0 \leq f+g \leq \kappa$ is polynomial.*

The proof of the above corallary results from an immediate application of the part (i) of proposition 3.

We have seen that we can allow some freedom on the hypothesis imposed on the elements of the class of KE-graphs; moreover, in the case of "bounded freedom", the relation between constructive and non-constructive versions remains invariant and polynomial. In the next paragraph we generalize this result in the case of WIS.

**The weighted case.** We have already seen that the KE-graphs are the ones for which the discrete duality gap $v(\mathrm{EC}(G))-\alpha(G)$ is equal to 0. Moreover, in the previous paragraph we have relaxed the condition $v(\mathrm{EC}(G))-\alpha(G)=0$, by allowing to this gap to be bounded above by a fixed positive constant. On the other hand, $v(\mathrm{IS}_r(G))-\alpha(G) \leq v(\mathrm{EC}(G))-\alpha(G)$; consequently, KE-graphs are the ones for which $v(\mathrm{IS}_r(G))-\alpha(G)=0$.

Of course, the question on the difference between the value of IS and the one of its linear relaxation can be also posed for WIS[8]. In [4], the authors introduce the class of graphs where $v(\mathrm{WIS}(G)) = v(\mathrm{WIS}_r(G))$ and call it the class of b-KE-graphs, for which they conceive an $O(n^{2.5})$ exact WIS-algorithm.

Here, we relax the condition $v(\mathrm{WIS}(G)) = v(\mathrm{WIS}_r(G))$ by a less restrictive one and we prove that, in the class of graphs defined by this relaxed information, WIS remains polynomial. Although the proof of the similar result in the non-weighted case resulted from a combinatorial interpretation leading to the consideration of an upper bound of the quantity $\alpha(G)-v(\mathrm{IS}_r(G))$, such a combinatorial interpretation for the weighted case, or, more generally, for the primal – dual approach, is much less natural. So, we give here a straightforward proof based upon linear programming arguments.

The purpose of this section is to prove the following result.

**Theorem 5.** *Consider the class of graphs $G=(V,E)$ satisfying $v(\mathrm{WIS}(G)) \geq v(\mathrm{WIS}_r(G))-\kappa$, where $\kappa$ is a fixed constant. Then, the problems: (i) decide if a graph $G$ belongs to this class and (ii) solve WIS in this class, are polynomial.*

[8] Let us note that the respective linear programs for $\mathrm{WIS}(G)$ and $\mathrm{WIS}_r(G)$ are identical to the ones for $\mathrm{IS}(G)$ and $\mathrm{IS}_r(G)$, up to the replacement of the unit-vector by a positive cost-vector with components $w_i$, $i=1,\ldots,n$.

It is well-known that the LP-relaxation $\mathrm{WIS}_r$ of WIS has the semi-integral property, i.e., each basic feasible optimal $\mathrm{WIS}_r$-solution $B$ assigns to the variables values drawn from the set $\{0, 1/2, 1\}$ (see [9], for a very interesting discussion about this fact). Starting from this property, it can be proved that if $V_1$ is the set of vertices corresponding to variables valued by 1 in $B$, then there exists a maximum independent set in $G$ that contains $V_1$ (in what follows, we denote by $V_1$, $V_{1/2}$ and $V_0$ the subsets of $V$ corresponding to LP-variables assigned by values 1, 1/2 and 0, respectively; of course, these three sets form a partition of $V$). Moreover, in [9], it is shown that $\mathrm{WIS}_r$ can be seen as the problem of computing the minimum edge covering in a bipartite graph. The key-operation of such a computation is the computation of a maximum matching, performed in $O(n^{2.5})$.

In lemma 6, we show how, given an instance $G = (V, E)$ of WIS, we can, either determine a partition of $V$ into sets $V_1$, $V_{1/2}$ and $V_0$, such that $V_{1/2} = \emptyset$ and so $\mathrm{WIS}(G)$ is solved to optimality, or we can reduce $G$ to a subgraph where the unique optimal solution for $\mathrm{WIS}_r$ is formed by assigning to all of its vertices the value 1/2. This result further extends the work of [9], where such a result does not appear, even if the possibility that 1/2 is assigned as unique value of all the vertices of a graph (when $\mathrm{WIS}_r$ is solved) is mentioned. More precisely, let us suppose the existence of a $\mathrm{WIS}_r$-solution $B$ assigning 1/2 to all the verices of an input graph $G = (V, E)$; this means that $V = V_{1/2}$. But, eventually, there exists another solution $B'$, of the same objective value as $B$, in which $V = V_1' \cup V_0' \cup V_{1/2}'$, with $V_1' \cup V_0' \neq \emptyset$. In lemma 6 we show that whenever the basic feasible optimal $\mathrm{WIS}_r$-solution $B$ assigns values 1/2 to all the variables, deciding if $B$ is unique, and if not, construct solution $B'$, is of polynomial complexity.

In what follows, when speaking of $\mathrm{WIS}_r$-solution, we refer to Nemhauser – Trotter's method ([9]).

**Lemma 6.** *Given a graph $G$, the $O(n^{4.5})$ procedure* **DECOMPOSITION** *determines a partition of $V$ into sets $V_1$, $V_{1/2}$ and $V_0$, and a solution for $\mathrm{WIS}_r(G)$ corresponding to this partition, such that either (i) $V_{1/2} = \emptyset$ and so $\mathrm{WIS}(G)$ is solved to optimality (by considering $v(\mathrm{WIS}(G)) = \sum_{v_i \in V_1} w_i$), or (ii) the unique optimal solution for $\mathrm{WIS}_r(G[V_{1/2}])$ is formed by assigning, to all of the vertices of $V_{1/2}$, the value 1/2.*

*Proof.* Since, by solving $\mathrm{WIS}_r$, $V_1$ is contained to at least a maximum independent set (and, consequently, $V_0$ to the minimum vertex covering associated with this independent set), in order to solve $\mathrm{WIS}(G)$ it suffices to solve $\mathrm{WIS}(G[V_{1/2}])$; then, $v(\mathrm{WIS}(G)) = v(\mathrm{WIS}(G[V_{1/2}])) + \sum_{v_i \in V_1} w_i$. So, in the case where $V_{1/2} = \emptyset$, $v(\mathrm{WIS}(G)) = \sum_{v_i \in V_1} w_i$, and this value is found in $O(n^{2.5})$ by solving $\mathrm{WIS}_r(G)$ (by the minimum edge covering computation of [9] mentionned above) and determining $V_1$.

The **repeat** loop of procedure **DECOMPOSITION** operates on $G[N]$, which verifies $v(\mathrm{WIS}(G[N])) = (1/2)\mathbf{w} \cdot \mathbf{1}_N$. So, if $\mathbf{w} \cdot (\mathbf{1}_{V_1'} + (1/2)\mathbf{1}_{V_{1/2}'} + \mathbf{1}_{\{v_0\}}) = (1/2)\mathbf{w} \cdot \mathbf{1}_N$, then the solution of $\mathrm{WIS}(G[N])$, consisting of assigning value 1 on the vertices of $V_1' \cup \{v_0\}$ and 1/2 on the ones of $V_{1/2}'$, is optimal; consequently,

**Procedure DECOMPOSITION**
**begin**
solve $\mathrm{WIS}_r(G)$ to obtain sets $V_0$, $V_1$, $V_{1/2}$;
stop $\leftarrow$ **false**;
**while** stop = **false and** $V_{1/2} \neq \emptyset$ **do**
$N' \leftarrow V_{1/2}$;
**repeat**
choose $v_0 \in N'$;
$N' \leftarrow N' \setminus \{v_0\}$;
$N \leftarrow V_{1/2} \setminus (\{v_0\} \cup \Gamma(v_0))$;
solve $\mathrm{WIS}_r(G[N])$;
{* $V'_0$, $V'_1$, $V'_{1/2}$ are the basic feasible sets of $V_{1/2} \setminus (\{v_0\} \cup \Gamma(v_0))$ *}
**until** $\mathbf{w} \cdot (\mathbf{1}_{V'_1} + (1/2)\mathbf{1}_{V'_{1/2}} + \mathbf{1}_{\{v_0\}}) = (1/2)\mathbf{w} \cdot \mathbf{1}_{V_{1/2}}$ **or** $N' = \emptyset$;
**if** $\mathbf{w} \cdot (\mathbf{1}_{V'_1} + (1/2)\mathbf{1}_{V'_{1/2}} + \mathbf{1}_{\{v_0\}}) = (1/2)\mathbf{w} \cdot \mathbf{1}_{V_{1/2}}$ **then**
$V_1 \leftarrow V_1 \cup V'_1 \cup \{v_0\}$;
$V_0 \leftarrow V'_0 \cup V_0$;
$V_{1/2} \leftarrow V'_{1/2}$;
**else** stop $\leftarrow$ **true**;
**fi od**
**end;**

the so-modified solution remains optimal for $\mathrm{WIS}_r(G)$ and, moreover, contains more components equal to 1 than the initial one. On the other hand, the **while** loop of procedure **DECOMPOSITION** converges, since at each iteration, $V_{1/2}$ decreases. Hence, if finally $V_{1/2} = \emptyset$, then procedure **DECOMPOSITION** outputs a 0-1 solution for $\mathrm{WIS}_r(G)$, therefore an optimal solution also for $\mathrm{WIS}(G)$.

In the opposite case, for every $v_0 \in V_{1/2}$, $v(\mathrm{WIS}_r(G[V_{1/2} \setminus (\{v_0\} \cup \Gamma(v_0))])) + \mathbf{1}_{\{v_0\}} \cdot \mathbf{w} < v(\mathrm{WIS}_r(G[V_{1/2}]))$, and this means that there is no optimal solution of $\mathrm{WIS}_r(G[V_{1/2}])$ assigning 1 to $v_0$. Since this is true for every $v_0 \in V_{1/2}$, the solution assigning 1/2 on all the vertices of $V_{1/2}$ is the unique optimal solution of $\mathrm{WIS}_r(G[V_{1/2}])$.

In the worst case, the solution of the linear program corresponding to $\mathrm{WIS}_r$ will be executed for every $v_0 \in V_{1/2}$ obtained on the first line of procedure **DECOMPOSITION**, the cardinality of the set $V_{1/2}$ decreasing after each execution; so, in the worst case, we will have $\sum_{i=1}^{n} i$ executions of the procedure solving $\mathrm{WIS}_r$ in $O(n^{2.5})$, hence a worst case complexity of $O(n^{4.5})$, q.e.d.

**Definition 7.** Let $\kappa$ be a fixed positive constant. The class $\kappa$-KE is defined as the class of graphs $G = (V, E)$, of order $n$, such that: (i) the solution of $\mathrm{WIS}_r(G)$ consisting of assigning value 1/2 on all of the vertices of $V$ is unique for $\mathrm{WIS}_r(G)$, and (ii) $(1/2)\mathbf{w} \cdot \mathbf{1}_n - \kappa \leq v(\mathrm{WIS}(G)) \leq (1/2)\mathbf{w} \cdot \mathbf{1}_n$.

We are now going to prove lemma 8 (the proof of which constitutes the main

**Algorithm STABLE**
**begin**
**if** $\kappa \leq 0$ **then call** the procedure of [4] to optimally solve WIS($G$);
**else**
choose $v_i v_j \in E$;
**for** $x \in \{i, j\}$ **do**
$V_x \leftarrow V \setminus \{v_x\}$;
$G_x \leftarrow G[V_x]$;
**call** procedure **DECOMPOSITION** on $G_x$;
{* $V_{x_1}$, $V_{x_{1/2}}$, $V_{x_0}$ the obtained sets *}
**if** $V_{x_{1/2}} = \emptyset$ **then** $S_x \leftarrow \emptyset$;
**else**
$V'_x \leftarrow V_{x_{1/2}}$;
$G'_x \leftarrow G[V'_x]$;
$\kappa \leftarrow \kappa - (1/2)$;
$S_x \leftarrow$ **STABLE**$(\kappa, G'_x)$;
**fi**
**od**
$\hat{x} \leftarrow \text{argmax}_{x \in \{i,j\}}\{\mathbf{w} \cdot (\mathbf{1}_{V_{x_1}} + \mathbf{1}_{S_x})\}$;
$S \leftarrow V_{\hat{x}_1} \cup S_{\hat{x}}$;
**fi**
**end.**

part of the proof of theorem 5), i.e., that WIS is polynomial for the class of $\kappa$-KE-graphs.

**Lemma 8.** *Let $\kappa$ be a fixed positive constant and let $G = (V, E)$ be a $\kappa$-KE-graph of order $n$. Algorithm* **STABLE** *determines a maximum-weight independent set for $G$ in $O(n^{4.5})$. Also,* **STABLE** *polynomially decides if a graph is $\kappa$-KE.*

*Proof.* Let us first prove the following emphasized proposition: *if there exists a maximum-weight independent set of $G$ not containing $v_x$, then $G'_x$, constructed by algorithm* **STABLE**, *is $[\kappa - (1/2)]$-KE.*

In fact, let us evaluate $v(\text{WIS}(G'_x))$. Let us first note that, from the hypothesis on $v_x$, $v(\text{WIS}(G_x)) = v(\text{WIS}(G))$. On the other hand, procedure **DECOMPOSITION**, called by algorithm **STABLE**, works in such a way that a maximum-weight independent set of $G_x$ is obtained by adding to a maximum-weight independent set of $G'_x$, the set $V_{x_1}$ (the vertices of $V_x$ valued by one in the basic feasible solution of $\text{WIS}_r(G_x)$); so,

$$\begin{aligned} v(\text{WIS}(G'_x)) &= v(\text{WIS}(G_x)) - \mathbf{1}_{V_{x_1}} \cdot \mathbf{w} \\ &= v(\text{WIS}(G)) - \mathbf{1}_{V_{x_1}} \cdot \mathbf{w} \\ &\geq (1/2)\mathbf{w} \cdot \mathbf{1}_n - \kappa - \mathbf{1}_{V_{x_1}} \cdot \mathbf{w} \end{aligned}$$

$$\geq (1/2)\mathbf{w}\cdot(\mathbf{1}_{\{v_x\}}+\mathbf{1}_{V_{x_1}}+\mathbf{1}_{V_{x_{1/2}}}+\mathbf{1}_{V_{x_0}})-\kappa-\mathbf{1}_{V_{x_1}}\cdot\mathbf{w}$$
$$\geq (1/2)\mathbf{1}_{V'_x}\cdot\mathbf{w}-(\kappa-(1/2)(\mathbf{1}_{\{v_x\}}+\mathbf{1}_{V_{x_0}}-\mathbf{1}_{V_{x_1}})\cdot\mathbf{w}) \qquad (1)$$

where the last inequality holds because $V'_x = V_{x_{1/2}}$.
By definition of the class $\kappa$-KE, the solution of $\mathrm{WIS}_r(G)$, consisting in assigning value $1/2$ on all of the vertices of $G$, is the unique optimal solution of the problem; so, since the solution determined by procedure **DECOMPOSITION** (called by **STABLE**) is feasible for $\mathrm{WIS}_r(G)$, we get

$$(\mathbf{1}_{V_{x_1}}+(1/2)\mathbf{1}_{V_{x_{1/2}}})\cdot\mathbf{w} < (1/2)\mathbf{w}\cdot(\mathbf{1}_{\{v_x\}}+\mathbf{1}_{V_{x_1}}+\mathbf{1}_{V_{x_{1/2}}}+\mathbf{1}_{V_{x_0}})$$

and since the above inequality takes place between two semi-integers we have

$$(\mathbf{1}_{V_{x_1}}+(1/2)\mathbf{1}_{V_{x_{1/2}}})\cdot\mathbf{w} \leq (1/2)\mathbf{w}\cdot(\mathbf{1}_{\{v_x\}}+\mathbf{1}_{V_{x_1}}+\mathbf{1}_{V_{x_{1/2}}}+\mathbf{1}_{V_{x_0}})-1/2$$

we so deduce that

$$(1/2)(\mathbf{1}_{\{v_x\}}+\mathbf{1}_{V_{x_0}}-\mathbf{1}_{V_{x_1}})\cdot\mathbf{w}\geq 1/2. \qquad (2)$$

From expressions (1) and (2), we get $v(\mathrm{WIS}(G'_x))\geq(1/2)\mathbf{1}_{V'_x}\cdot\mathbf{w}-[\kappa-(1/2)]$.
Finally, let us recall that procedure **DECOMPOSITION** is conceived in such a way that the solution assigning $1/2$ on all of the vertices of $G'_x$ is the unique optimal solution of $\mathrm{WIS}_r(G'_x)$; this concludes the proof of the emphasized proposition.

The recursive algorithm **STABLE** explores a binary tree of depth $2\kappa$, the nodes of which correspond to the edges $v_iv_j$ chosen at the first line of the **else** clause of the outer **if** condition. If there exists a branch along which $v_x$ verifies the hypothesis of the emphasized proposition, then from its conclusion, and by an immediate induction, the graph, obtained by algorithm **STABLE**, corresponding to the leaf of this branch is **b**-KE, so, WIS is polynomial for this graph ([4]). Observe finally that, for all $k$, if we denote by $G^{(k)}$ the graph treated during the $k$th recursive call of algorithm **STABLE**, the sum of a maximum-weight independent set of ${G'_x}^{(k)}$ with the weight of the set $V_{x_1}^{(k)}$, constitutes a maximum-weight independent set for ${G'_x}^{(k)}$, hence for $G^{(k)}$ (because of the hypothesis that there exists at least a maximum-weight independent set of $G^{(k)}$ not containing $v_x$).
On the other hand, for every edge $v_iv_j$ of $G^{(k)}$, both of its endpoints cannot simultaneously belong to every independent set (a fortiori optimal), this fact guaranteeing that such a branch (leading to a **b**-KE-graph) always exists; so, algorithm **STABLE**, which, in fact, entirely explores this binary tree by choosing among the independent sets corresponding to the branches, the one of maximum weight, it determines a maximum-weight independent set of the initial graph (recall that the algorithm of [4], called here on the leaves of the binary tree, returns an empty solution if an input graph is not **b**-KE).

So, algorithm **STABLE** is called, at worst, on all of the nodes of a fictive binary tree of depth $2\kappa$ and order $4^\kappa$ (a constant since $\kappa$ is a fixed constant), the dominating operation of each call being procedure **DECOMPOSITION** of complexity $O(n^{4.5})$; consequently, the overall algorithm's complexity

is of $O(n^{4.5})$.
Finally, remark that **STABLE** can be trivially used to decide if, for a given (fixed constant) $\kappa$, a graph is $\kappa$-KE.

Let us remark that in the light of the previous result the complexity of the algorithms of proposition 3 is also of $O(n^{4.5})$.

We are now well-prepared to conclude the proof of theorem 5.

Consider a constant $\kappa$ and a graph $G = (V, E)$ such that $v(\mathrm{WIS}(G)) \geq v(\mathrm{WIS}_r(G)) - \kappa$. Procedure **DECOMPOSITION** allows to partition $V$ into three sets $V_0$, $V_1$ and $V_{1/2}$ in such a way that $v(\mathrm{WIS}(G)) = v(\mathrm{WIS}(G[V_{1/2}])) + \mathbf{w} \cdot \mathbf{1}_{V_1}$ and, furthermore, $v(\mathrm{WIS}_r(G)) = v(\mathrm{WIS}_r(G[V_{1/2}])) + \mathbf{w} \cdot \mathbf{1}_{V_1}$. Moreover, in order to construct an optimal solution of $\mathrm{WIS}(G)$, one has simply to complete an optimal solution of $\mathrm{WIS}(G[V_{1/2}])$ with the vertices of $V_1$. The above expressions allow to establish that $v(\mathrm{WIS}(G[V_{1/2}])) \geq v(\mathrm{WIS}_r(G[V_{1/2}])) - \kappa$; furthermore, using lemma 6 one can be sure that the graph $G[V_{1/2}]$ is $\kappa$-KE. Lemma 8 concludes then the proof of theorem 5.

### 4.3 Problems $\mathrm{S}_\kappa$

The classes considered in sections 4.1 and 4.2 were restrictive enough to allow facility for the constructive framework; so, for these classes, the constructive and non-constructive points of view are of identical facilities and one cannot dissociate them.

In order to capture the boundary between the two frameworks, let us now consider, starting from IS, the transformation consisting in integrating in the instance a vaguer information on the optimal value, i.e., consider only the order of the optimal value with respect to input-size $n$. This thought process leads us to the following problems introduced in [6].

For every constant $\kappa > 1$, we define the stability problem $\mathrm{S}_\kappa$ corresponding to the restriction of IS to graphs (of order $n$) admitting stability number greater than or equal to $n/\kappa$; we also define the problem $\mathrm{S}_{\kappa^\infty}$, an instance of which is the pair $(G, \kappa)$, where $\kappa \geq 1$ and $G$ is a graph of order $n$ with $\alpha(G) \geq n/\kappa$. Here also, the objective is to determine a maximum independent set of $G$.

The first remark concerning these problems is that the recognition of their instances is NP-complete for $\kappa \geq 2$ (contrarily to the problems considered in sections 4.1 and 4.2).

**Proposition 9.** *The problem of deciding if, in graph $G$ of order $n$, $\alpha(G) \geq n/\kappa$ is NP-complete $\forall \kappa \geq 2$, $\kappa \in \mathbb{N}$.*

*Proof.* The reduction is from IS, where we have to decide if, for a graph $G$, $\alpha(G) \geq \ell$, for a constant $\ell \in \mathbb{N}$.

*Remark.* We can consider that $\ell \leq n/\kappa$.

If not ($\ell > n/\kappa$), we add a clique $K_c$ in $G$ and we insert all edges between vertices of $G$ and vertices of $K_c$. The new graph $G'$ is of order $n+c$ vertices and, moreover, $\alpha(G) = \alpha(G')$. If $c \geq \kappa\ell - n$, then $\ell \leq (n + c)/\kappa$.

*Remark.* The quantities $n$ and $\ell$ are multiples of $\kappa - 1$.

If not, we consider graph $G'$ consisting of $\kappa - 1$ disjoint copies of $G$. Its order is now $(\kappa - 1)n$ and the question becomes if $\alpha(G') \geq \ell(\kappa - 1)$.

*Remark.* Finally, $\ell = n/\kappa$.

If $\ell \neq \kappa$, we add $\sigma$ independent vertices in $G$. The resulting graph $G'$ has $n + \sigma$ vertices and $\alpha(G') = \alpha(G) + \sigma$. So, we have to decide if $\alpha(G) + \sigma \geq \ell + \sigma$. The fact $\ell + \sigma = (n + \sigma)/\kappa$ can be assured if $\sigma = [n/(\kappa - 1)] - [\kappa\ell/(\kappa - 1)]$. The two first remarks assure $\sigma > \mathbb{N}^+$.

The combination of the three above remarks conclude the NP-completeness claimed.

The above result indicates that the information integrated in the instance of the problems $S_\kappa$ is quite strong. Consequently, it is reasonable to ask ourselves if one can exploit this information for approximating these problems.

We then have the following result.

**Theorem 10.** *For every constant $\rho$, there is no polynomial time approximation algorithm for $S_{\kappa^\infty}$ which guarantees approximation ratio greater than or equal to $\rho$.*

In other words, $S_{\kappa^\infty}$ does not admit constant-ratio polynomial time approximation algorithm. For the problems $S_\kappa$, this means that, *if there exists a polynomial time approximation algorithm guaranteeing, for every problem $S_\kappa$, an approximation ratio $\rho(\kappa)$, then the mapping $\kappa \mapsto \rho(\kappa)$ tends to 0 whenever $\kappa \to \infty$.*

In fact, we can even precise the above remark by computing the convergence velocity of $\rho$ to 0.

**Theorem 11.** *$\exists \varepsilon > 0$ and $\exists \kappa_0$ such that $\forall \kappa \geq \kappa_0$, there does not exist algorithm polynomial in $n$ (but, eventually, exponential in $\kappa$) for $S_\kappa$ guaranteeing approximation ratio $(1/\kappa)^\varepsilon$.*

*Proof.* In [1], it is proved that unless P = NP, IS, even for above-bounded-maximum-degree graphs (let us denote the class of IS defined on such graphs by B($\Delta$)S), cannot be approximated by a polynomial time approximation schema[9] (PTAS).

Using the result of [1], we prove the following intermediate result (lemma 12).

**Lemma 12.** *$\exists \kappa_0$ such that $S_{\kappa_0}$ does not admit a PTAS.*

*Proof.* **(Lemma 12.)** Let us prove that if, $\forall \kappa$, $S_\kappa$ admits a PTAS, then B($\Delta$)S also admits a PTAS.
Recall that in a graph $G$ of order $n$, every maximal independent set is greater than or equal to $n/(\Delta + 1)$; consequently, $\alpha(G) \geq n/(\Delta + 1)$ and, for each $\Delta$, problem B($\Delta$)S is a sub-problem of $S_{\Delta+1}$, solved by a PTAS on the hypothesis that, $\forall \kappa$, $S_\kappa$ can be solved by a PTAS, and this completes the proof of lemma 12.

[9] A sequence of polynomial time approximation algorithms, indexed by $\epsilon$, guaranteeing, for every $\epsilon$, approximation ratio of $1 - \epsilon$.

Let now $G$ be an instance of $S_{\kappa_0}$ of size $n$. For all integers $m$, consider the graph $G^m$ defined in the proof of proposition 1 and let $n_m$ be its order; then, $n_m = n^m$. The properties of this construction allow to establish that, if $G$ is an instance of $S_\kappa$, then $G^m$ is an instance of $S_{\kappa^m}$.
Suppose now that, $\forall \varepsilon > 0$, $\exists \kappa > \kappa_0$ such that $S_\kappa$ is $(1/\kappa)^\varepsilon$-approximable. For $\eta > 0$, choose $\varepsilon > 0$ such that $(1/\kappa_0^{3/2})^\varepsilon > 1-\eta$ and $\kappa > \kappa_0$ such that $S_\kappa$ is $(1/\kappa)^\varepsilon$-approximable. We then determine integer $m$ such that $\kappa_0^m \leq \kappa < \kappa_0^{m+1}$; $G^m$ is an instance of $S_\kappa$, and so, the algorithm supposed to solve it allows to determine in polynomial time an independent set of cardinality $\alpha'(G^m) \geq (1/\kappa)^\varepsilon)\alpha(G^m) \geq (1/\kappa_0^{1+m})^\varepsilon)\alpha(G^m)$.
Following the above discussion, we can deduce an independent set of cardinality $\alpha'(G^m)^{1/m} \geq (1/\kappa_0^{(1+m)/m})^\varepsilon)(\alpha(G^m))^{1/m} \geq (1/\kappa_0^{3/2})^\varepsilon)\alpha(G) \geq (1-\eta)\alpha(G)$ for $G$. This holding, $\forall \eta \in ]0,1[$, we have determined a PTAS for $S_\kappa$, a contradiction. This completes the proof of theorem 11

We cannot yet characterize the hardness of approximating problems $S_\kappa$ for a fixed $\kappa$. The only remark we can make is that, for $\kappa > 2$, $S_\kappa$ is polynomially constant-approximable.

In fact, let us consider an instance $G$ of $S_{2-\epsilon}$ for a positive constant $\epsilon$. Then, $\alpha(G) \geq n/(2-\epsilon) \geq [(1/2)+\epsilon']n$ for $\epsilon' > 0$. Let us denote by $e$ the number of vertices exposed[10] with respect to a maximum matching $M$ of cardinality $m$. We have $n = 2m + e$ and $\alpha(G) \leq m + \epsilon$. If we choose as approximate solution for IS the independent set formed by these vertices, the previous expressions lead to $e \geq 2\epsilon' n$ and the constant ratio $2\epsilon'$ is guaranteed for $S_{2-\epsilon}$.

The approximability of problems $S_\kappa$ is particularly interesting. In fact, if one could precise (see theorem 11) a threshold $\kappa_0$ such that, $\forall \kappa \geq \kappa_0$, $S_\kappa$ is not constant-approximable, then one would bring to the fore a problem constant-approximable in the non-constructive framework within a ratio $1/\kappa$ which, at the same time, is not constant-approximable in the constructive framework.

Furthermore, the approximability of problems $S_\kappa$ is strongly linked to the approximability of other combinatorial problems, for example, convexe programming problems, including quadratic programming as sub-case, or, even, to the open problem of improving the approximation ratio 2 for minimum vertex covering.

More precisely, the following conditional results are proved in [6].

**Theorem 13.** ([6]). *Let $\rho < 1$ be a fixed positive constant. Under the hypothesis $P \neq NP$, (i) the non-existence of a $\rho$-approximation polynomial time algorithm for $S_3$ implies that no polynomial time approximation algorithm for VC can guarantee an approximation ratio strictly smaller than 3/2; (ii) if, on the contrary, such an algorithm exists for $S_3$, then there exists an algorithm[11] for VC guaranteeing an approximation ratio smaller than, or equal to, $2 - (\rho/6) < 2$.*

[10] The set of these vertices is non-empty because of the hypothesis $\alpha(G) > n/2$.

[11] This algorithm is constructive, since the proof of part (ii) of theorem 13 is entirely constructive.

In the case $\kappa = 2$, we can obtain the following result.

**Proposition 14.** ([6]). *On the hypothesis $P \neq NP$, if $S_2$ is non-constant approximable, then no polynomial time approximation algorithm for VC can guarantee an approximation ratio $\rho < 2 - \epsilon$, for a fixed positive constant $\epsilon$.*

The (maximization) convex programming problem considered in [6] is defined as follows:

$$\mathrm{CPM}(\kappa) \;=\; \begin{cases} \max \displaystyle\sum_{i \in \{1,\ldots,n\}} f(x_i) \\ x \in \mathcal{P} \end{cases}$$

where $\mathcal{P}$ is a polytope defined by a finite number of constraints, and $f$ belongs to a family $\mathcal{F}$ of functions increasing in $[0,1]$, with $f(0) = 0$ for every $f \in \mathcal{F}$, verifying the property

$$\inf_{f \in \mathcal{F}} \left\{ \frac{f(\frac{1}{2})}{f(1)} \right\} \in \left[0, \frac{1}{\kappa}\right[, \quad \kappa \in \mathbb{R} \setminus [0,2[. \tag{3}$$

We have then the following result.

**Theorem 15.** ([6]). *If there exists $\kappa \in \mathbb{R} \setminus [0,2[$ such that $S_\kappa$ does not admit a polynomial time algorithm guaranteeing a maximal independent set greater than $\rho n$ for a fixed positive constant $\rho < 1$, then there does not exist a polynomial time approximation algorithm for CPM($\kappa$) guaranteeing an approximation ratio greater than $\kappa \inf_{f \in \mathcal{F}} \{f(1/2)/f(1)\}$.*

By applying the result of theorem 15 on particular families $\mathcal{F}$ ([6]), we can deduce the following corollaries.

**Corollary 16.** ([6]). *If $S_3$ does not admit a polynomial time approximation algorithm of (universally) constant ratio, then the quadratic programming problem does not admit a polynomial time approximation algorithm guaranteeing an approximation ratio greater than or equal to $3/4$, unless $P = NP$.*

**Corollary 17.** ([6]). *If there exists $\kappa$ such that $S_\kappa$ does not admit a polynomial time algorithm guaranteeing a maximal independent set greater than $\rho n$ for a fixed positive constant $\rho < 1$, then the problem of maximizing a convex function in a polytope does not admit a constant-ratio polynomial time approximation algorithm.*

Analogous results are given in [6] for concave programming problems.

A slightly different problem, with respect to the approximability of $\mathrm{S}_\kappa$, is defined as follows: *given an instance $G$ of $\mathrm{S}_\kappa$, determine an independent set of size $n/\kappa$.* This is a purely constructive problem, harder than the approximability of $\mathrm{S}_\kappa$ with ratio $1/\kappa$.

For this new problem, starting from proposition 9, we can immediately deduce the following result.

**Proposition 18. (Constructive version of proposition 9.)**
*If $P \neq NP$, then $\forall \kappa \geq 3$, $\kappa \in \mathbb{N}$, there cannot exist a pair of $(\mathcal{A}, f)$, where $\mathcal{A}$ is an algorithm and $f$ a function, both ($\mathcal{A}$ and $f$) polynomial in $n$ (but, eventually, exponential in $\kappa$) such that, for all graphs $G$ of order $n$ instances of $S_\kappa$, $\mathcal{A}$ determines an independent set greater than, or equal to $n/\kappa$ with complexity less than, or equal to $f(n)$.*

The proof is an easy conjugation of the ideas of theorem 2 to the result of proposition 9.

## References


1. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and intractability of approximation problems. Proc. FOCS (1992) 14-23
2. Ausiello, G., Crescenzi, P., Protasi, M.: Approximate solution of NP Optimization Problems. Theoretical Computer Science (to appear)
3. Berge, C.: Graphs and hypergraphs. North Holland (1973)
4. Bourjolly, J. - M., Hammer, P. L., Simeone, B.: Node-weighted graphs having the König-Egervary Property. Math. Prog. Study **22** (1984) 44-63
5. Demange, M., Paschos, V. Th.: On an approximation measure founded on the links between optimization and polynomial approximation theory. Theoretical Computer Science (to appear)
6. Demange, M., Paschos, V. Th.: The approximability behaviour of some combinatorial problems with respect to the approximability of a class of maximum independent set problems. Computational Optimization and Applications (to appear)
7. Deming, R. W.: Independence numbers of graphs – an extension of the König - Egervary theorem. Disc. Math. **27** (1979) 23-33
8. Garey, M. R., Johnson, D. S.: Computers and intractability. A guide to the theory of NP-completeness. Freeman (1979)
9. Nemhauser, G. L. Trotter, Jr, L. E.: Vertex packings: structural properties and algorithms. Math. Prog. **8** (1975) 232-248