Yugoslav Journal of Operations Research
13 (2003), Number 1, 3-24

REDUCING OFF-LINE TO ON-LINE: AN EXAMPLE AND ITS APPLICATIONS

Marc DEMANGE

*ESSEC Business School**
Cergy-Pontoise Cedex, France
demange@essec.fr

Abstract: We study on-line versions of maximum weighted hereditary subgraph problems for which the instance is revealed in two clusters. We focus on the comparison of these on-line problems with their respective off-line versions. In [3], we have reduced on-line versions to the off-line ones in order to devise competitive analysis for such problems. In this paper, we first devise hardness results pointing out that this previous analysis was tight. Then, we propose a process that allows, for a large class of hereditary problems, to transform an on-line algorithm into an off-line one with improvement of the guarantees. This result can be seen as an inverse version of our previous work. It brings to the fore a hardness gap between on-line and off-line versions of those problems. This result does not apply in the case of maximizing a k-colorable induced subgraph of a given graph. For this problem we point out that, contrary to the first case, the on-line version is almost as well approximated as the off-line one.

Keywords: Combinatorial problems, on-line computation, reductions, hereditary subgraph problem.

1. CONTEXT AND AIMS OF THE PAPER

A set-property π, assigning to every finite set V a Boolean value (either *true* if V satisfies π, or *false* in the opposite case), is hereditary if, whenever V satisfies π, so does every subset of V. π is called *trivial* if it is satisfied for only a finite number of sets, or unsatisfied for only a finite number of sets. Heredity is very natural in operations research; a generic example is the case where constraints represent the saturation of a shared resource: it is quite natural that a part of a feasible program

---

* also at CERMSEM, Université Paris I, Paris, France.

remains feasible ("who can do more can do less"). Consider then a general decision problem where one has to optimally select (accept) a subset V' of alternatives among V. Assume then that profits are associated to alternatives: to select $a \in V$ induces a profit $p(a)$ while to reject it induces neither a cost, nor a profit. The related problem is then to find a feasible subset $V' \subseteq V$ maximizing the additive profit $p(V') = \sum_{a \in V'} p(a)$. Whenever feasibility represents the non-saturation of resources, it is a hereditary property. We also suppose that it is not trivial and that it can be polynomially tested for every input set. Many times, such a set property can be defined by a graph property since, given a graph $G = (V, E)$, subsets of V are one-to-one associated to induced subgraphs of G. In this context, the problem of finding in V a feasible set V' maximizing the additive profit is an instance of the combinatorial problem (or class of problems) WHG, called maximum hereditary subgraph problem. The unweighted problem HG corresponds to the case where all profits coincide. Two typical examples of such problems are maximum weighted independent set and maximum weighted clique problems. In our generic model, both problems correspond to the case of pair wise incompatibilities: for the maximum independent set problem, edges of the input graph represent incompatibilities whereas, for the maximum clique one, the input is a compatibility graph. HG and WHG are NP-hard [5] and moreover are hard to approximate [7, 8].

In *on-line* computation, the instance is not supposed to be completely known before one begins to solve it, but it is revealed step-by-step. At each step a new part of the instance is revealed and one has to decide how this part contributes to the solution under construction. Given an integer t, we denote by L(W)HG (t) the on-line version (or on-line model) of (W)HG where the instance is revealed in t steps. At each step, a subgraph of new vertices is revealed together with edges linking new vertices and already known vertices. An on-line algorithm for this model selects, at each step, some vertices that will be included in the solution. The set of so selected vertices has to satisfy π. The quality of an on-line algorithm is measured, for every instance, by the ratio of the value of the on-line solution to the optimal value of the whole instance. The algorithm is said to guarantee *competitivity ratio* ρ *against optimality* (where $0 < \rho \leq 1$ for the case of maximization) if, for every one-line instance, the related ratio is at least ρ.

In [3], we have studied LHG (t) for different values of t and also for $t = n$, corresponding to the case where V is revealed vertex by vertex. In this paper we have also considered LWHG(2) we are interested in. It has its own interest with regard to the application fields: suppose for instance that vertices do not appear at the same time, one can defer the decision in order to take into account new information, but one has imperatively to take a first decision at a fixed deadline for the already known vertices. It corresponds to on-line problem LWHG(2) where the first part of the instance consists of all vertices that are known (submitted) before the deadline, and the second one contains the last vertices. From a theoretical point of view, this on-line version is particularly interesting: it corresponds to the first level of on-line framework for this problem (LWHG(1) = WHG is the usual off-line problem). Consequently, this on-line model is really suitable for understanding in what measure the on-line framework influences the hardness of the problem.

The aim of this note is namely to study the borderline between on-line and off-line, and more precisely, to evaluate the hardness gap between WHG and LWHG(2). For the case of WHG, seen as LWHG(1), the competitivity ratio corresponds exactly to the usual notion of approximation ratio ([5]). So, approximating the off-line problem with performance guarantees and solving the on-line version within a competitivity ratio are very similar points of view that we try to compare. The extra difficulty of LWHG(2) is due to the fact that a good choice of vertices during the first step can drastically restrict the possibilities during the second step (for preserving feasibility); so, it can be a very bad choice for the whole instance.

Approximation preserving reductions, linking the approximation behaviour of (even really) different problems, are very useful in order to compare the approximation hardness of those problems. Such reductions describe process allowing us to transfer an approximation result for the former problem into another approximation result for the latter. The *expansion* of the reduction is a function describing how the approximation ratio is affected during the transfer. In order to link the approximation behaviour of WHG to the competitivity behaviour of LWHG(2), we conceive reductions that are able to link an off-line problem and an on-line one. This generalization of approximation preserving reductions already appears as very interesting and useful in on-line framework. In [3], we have presented a first example of such a reduction that transforms an approximation algorithm for WHG into an on-line algorithm for LWHG(2). We have also given other similar examples allowing devising competitivity analysis of several on-line versions of HG. The expansion of the reduction from WHG to LWHG(2) can be considered as a first evaluation of the relative hardness of both problems. In what follows, we show that, in a way, this first evaluation is tight.

We first devise a hardness result for LWHG(2). We deduce that the above reduction cannot be significantly improved: in the other case, an optimal algorithm for WHG could be transformed into an on-line algorithm for LWHG(2), contradicting our hardness result. But it does not allow us to compare polynomial-time approximation of WHG and polynomial-time on-line solution of LWHG(2). Our main result is then a reduction that allows us to transfer, for a class of hereditary properties, an on-line algorithm for LWHG(2) into an approximation algorithm for WHG with improved ratio. This reduction holds either for polynomial, or for non-polynomial algorithms. It allows us to devise hardness results for polynomial-time on-line algorithms. It also points out that improving the on-line algorithm given in [3] for LWHG(2) would allow us to improve the best known polynomial-time approximation of WHG.

From a theoretical point of view, we find this result to be interesting for two reasons. It brings to the fore a hardness gap between an off-line problem and its on-line version. It also allows us to achieve hardness results dealing with polynomial-time on-line algorithms, whereas most of on-line hardness results do not take into account the completion time. But the algorithmic complexity is precisely a significant parameter in the framework of on-line models for which the instance is revealed per large clusters. The second interest is that this is, to our knowledge, the first non-trivial reduction that exploits an on-line algorithm in order to solve an off-line problem. In [3], we have already devised reductions allowing changing off-line algorithms into on-line ones and also reductions between on-line problems. So, in this paper we give an example of the third possible case of reductions in on-line context.

Finally, in the last section we focus on maximum k-colorable induced subgraph problem for some $k \geq 2$. The previous result does not apply for this case; we show that its on-line version is almost as well approximated as the off-line version.

2. DEFINITIONS AND NOTATIONS

We denote by $\mathbb{N}$ the set of positive integers and by $\mathbb{Q}$ the set of rational numbers. For a positive real number x, $\lfloor x \rfloor$ denotes the largest integer less than, or equal to x, and $\lceil x \rceil$ denotes the smallest integer strictly greater than x. In particular, if x is an integer, $x = \lfloor x \rfloor = \lceil x \rceil - 1$. For a rational number r, we define its *dimension* by the minimum value of pq where p, q are integers such that $r = p/q$. For a real vector w, we denote by $|w|$ its L^1-norm; if E is a finite set, $|E|$ denotes its cardinality (the L^1-norm of its characteristic vector).

In this work, we will only consider simple graphs ([1]), i.e., non-oriented, without loop and with at most one edge between every two vertices. Let $G = (V, E)$ be a graph, we denote by $n(G)$ (or n) its order $(n = |V|)$. For every set of vertices $V' \subset V$, we denote by $G[V']$ the subgraph of G induced by V'. Let us then assign to every vertex v a rational weight w_v; we denote by w the vector of weights (each component is associated to a vertex and corresponds to its rational weight); for a set of vertices $V' \subset V$, its weight is defined by $w(V') = \sum_{v \in V'} w_v$; $w(V')$ is also called the weight of the graph $G[V']$. We also denote by $W = w(V)$ the weight of the whole graph. (G, w) is called a weighted graph; $\mathcal{G}$ denotes the set of finite graphs and $\mathcal{G}_w$ the set of finite weighted graphs.

2.1 Hereditary properties

Definition 1. Hereditary property
Let $\pi : \mathcal{G} \to \{false, true\}$ be a graph-property.
(i) π is hereditary if:

$$\forall G = (V, E) \in \mathcal{G},\ \pi(G) \Rightarrow \forall V' \subset V,\ \pi(G[V'])$$

(ii) π is trivial if it is satisfied for only a finite number of graphs, or is unsatisfied for only a finite number of graphs.

The following remark is immediately deduced from the definition:

Remark 1. Let π be a non-trivial hereditary graph-property:
(i) $\forall n \in \mathbb{N},\ n \neq 0$, there exists a graph of order n satisfying π.
(ii) $\exists K \in \mathbb{N},\ \forall n \geq K$, there exists a graph of order n that does not satisfy π.

In this work, we consider a polynomially computable non-trivial property π. We assume without loss of generality that a single set, seen as a graph $(\{v\},\emptyset)$, satisfies π; in the other case, x would never belong to a feasible set and could also be drawn out of the instance. Let $G=(V,E)$ be a graph, a subgraph $G[V']$, $V'\subset V$, $V'\neq V$, satisfying π is called *maximal* (for inclusion), or non-extendible if, $\forall v\in V\setminus V'$, $G[V'\cup\{v\}]$ does not satisfy π. An independent set is a graph without edges and a clique is a complete graph. "*Independent set*" and "*clique*" are two well-known hereditary graph properties that play a specific rule in what follows. In both cases the following properties **C1** and **C2** can be immediately deduced:

C1. For every graph $G_1=(V_1,E_1)$ and every size n_2, there exists a graph $G=(V,E)$ such that $V=V_1\cup V_2$, $|V_2|=n_2$, $G[V_1]=G_1$, V_2 satisfies π and, $\forall(v_1,v_2)\in V_1\times V_2$, $\{v_1,v_2\}$ does not satisfy π.
C2. For every graph $G_1=(V_1,E_1)$ and every size n_2, there exists a graph $G=(V,E)$ such that $V=V_1\cup V_2$, $|V_2|=n_2$, $G[V_1]=G_1$ and every single set $\{x_2\}\subset V_2$ is a maximal (for inclusion) set satisfying π in G.

One can easily show that π is unsatisfied for exactly one graph of order 2 if and only if it is either "*independent set*" or "*clique*". We will also consider examples of hereditary properties that are satisfied for every graph of order 2. In order to point out, in a more general framework, properties that look like **C1** and **C2**, we introduce the following definition:

Definition 2. *Let π be a hereditary graph property and let k be an integer.*
(i) We say that π satisfies the k-boundary condition if, for every $n\geq k+1$, there exists a graph of order n such that every induced subgraph of G of order $k+1$ does not satisfy π.
(ii) We say that π satisfies the k-star-boundary condition if every graph of maximum degree at least k (containing a star of size $k+1$ us partial subgraph) does not satisfy π.

Proposition 1. *Let π be a hereditary graph property that is satisfied for every single vertex. π satisfies a k-boundary condition, for some k, if and only if it is false for some clique or independent set.*

Proof: Let us first suppose that a graph $H=(V_H,E_H)$, that is either a clique or an independent set, does not satisfy π. Let us define $k=|V_H|-1$ (H is not a single set since it does not satisfy π). Then, for every $n\geq k+1$, there exists a graph of order n (a clique or an independent set, respectively) which every subgraph of order $k+1$ is isomorph to H; so, π satisfies the k-boundary-condition.

Let us now suppose that every independent set and every clique satisfy π. For $(m,n)\in\mathbb{N}\times\mathbb{N}$, there is (see for instance [1]) a finite number $R(m,n)$ (the so-called Ramsey number) so that every graph of order at least $R(m,n)$ contains either a clique of size m or an independent set of size n. Consequently, for every integer k, every graph of order at least $R(k+1,k+1)$ contains an induced subgraph of order $k+1$ that satisfies π, which concludes the proof. ♦

This proposition brings to the fore that many hereditary graph properties satisfy a k-boundary condition. Among over let us mention independent, clique, planar, acyclic, k-colorable, of maximum degree k, with at most $k(k+1)/2$-edges satisfying, respectively, $1-, 1-, 4-, 2-, k-, k+1-$ and k-boundary-condition. k-star-boundary condition trivially implies k-boundary condition. l-colorable with $l \geq 2$ does not satisfy the k-star-boundary condition for any value of k; on the other hand, properties "of maximum degree k" and "with at most k edges" satisfy $k+1$-star-boundary condition. Let us also note that, for every $l \geq k$, k - (star)-boundary condition implies l - (star)-boundary condition. The following properties are natural extensions of **C1** and **C2**:

C1.1$_k$. For every graph $G_1 = (V_1, E_1)$ of order $n_1 \geq k$ and every integer n_2, there exists a graph $G = (V, E)$ such that $V = V_1 \cup V_2, |V_2| = n_2, G[V_1] = G_1$, V_2 satisfies π and, $\forall V_1' \subseteq V_1, |V_1'| = k, \forall v_2 \in V_2, V_1' \cup \{v_2\}$ does not satisfy π.

C1.2$_k$. For every graph $G_1 = (V_1, E_1)$ and every integer $n_2 \geq k$, there exists a graph $G = (V, E)$ such that $V = V_1 \cup V_2, |V_2| = n_2, G[V_1] = G_1$, V_2 satisfies π, and, $\forall V_2' \subseteq V_2$, $|V_2'| = k, \forall v_1 \in V_1, V_2' \cup \{v_1\}$ does not satisfy π.

C2$_k$. For every graph $G_1 = (V_1, E_1)$ and every integer $n_2 \geq k$, there exists a graph $G = (V, E)$ such that $V = V_1 \cup V_2, |V_2| = n_2, G[V_1] = G_1$ and, $\forall V_2' \subseteq V_2, |V_2'| = k, \forall v \in V \setminus V_2'$, $V_2' \cup \{v\}$ does not satisfy π.

C1 and **C2** correspond to the case where $k = 1$. **C1.1**$_k$ and **C1.2**$_k$ trivially hold if π satisfies the k-star-boundary condition, and **C2**$_k$ is satisfied if π satisfies the k-boundary condition. In fact, let us suppose that k-boundary condition holds for a given k; then, given a graph $G_1 = (V_1, E_1)$ and an integer $n_2 \geq k$, let H be a graph of order $n_2 + 1$ which every induced subgraph of order $k+1$ does not satisfy π; let also v_0 be a vertex of $H = (V_H, E_H)$ and let $G_2 = H[V_H \setminus \{v_0\}] = (V_2, E_2)$. We define $G = (V_1 \cup V_2, E)$ such that $G[V_1] = G_1$, $G[V_2] = G_2$ and for every $(v_1, v_2) \in V_1 \times V_2$, v_1 and v_2 are linked by an edge if and only if $(v_0 v_2) \in E_H$. Then, $\forall V_2' \subseteq V_2, |V_2'| = k, \forall v \in V \setminus V_2', G[V_2' \cup \{v\}]$ is a subgraph of H of order $k+1$; so it does not satisfy π, that corresponds to condition **C2**$_k$.

Property **C1** has a nice consequence, called **C3** that will be useful in the sequel:

C3. Let $G_1 = (V_1, E_1)$ be a graph, let $V_1' \subseteq V_1$, satisfying π, and let n_2 be an integer, there exists a graph $G = (V, E)$ such that $V = V_1 \cup V_2, |V_2| = n_2, G[V_1] = G_1$, V_2 satisfies π and V_1' is a maximal subgraph of $G[V_1' \cup V_2]$ satisfying π.

C1.1$_k$ corresponds to property **C3** for every V_1' of size k. If $G[V_1']$ has at least one edge and π is "*without triangles*", then **C3** is also satisfied. Let us finally point out another situation for which property **C3** holds:

Proposition 2. *Let π be a hereditary property, let $G_1 = (V_1, E_1)$ be a graph, and let $V_1' \neq V_1$, be such that $G[V_1']$ is a maximal induced subgraph of G satisfying π. Then property* **C3** *holds.*

Proof: Let $v_1 \in V_1 \setminus V_1'$, $G_1[V_1' \cup \{v_1\}]$ does not satisfy π. Let n_2 be an integer and $G_2 = (V_2, E_2)$ be a graph of order n_2 satisfying π. Then, we define edges between V_1 and V_2 such that every vertex of V_2 has the same neighbourhood in V_1 as v_1. Consequently, $\forall v_2 \in V_2$, $G[V_1' \cup \{v_2\}] = G_1[V_1' \cup \{v_1\}]$ does not satisfy π.

2.2. WHG and approximation algorithms

WHG is the problem of finding, for every weighted graph (G,w), a maximum weight induced subgraph of G satisfying π. Let A be a polynomial-time algorithm for WHG computing, for every weighted graph (G,w), an induced subgraph of G satisfying π. We denote by $A(G,w)$ the subgraph computed by A (or equivalently its vertex set) and by $\lambda_A(G,w)$ the weight of $A(G,w)$; we also denote by $\beta(G,w)$ the optimal value of instance (G,w), i.e., the maximum weight of an induced subgraph of G satisfying π. Several particular cases of WHG are well-known; let us notably mention the maximum weighted independent set problem denoted by WS, the maximum weighted clique problem denoted by WK, and the maximum weighted k-colorable subgraph denoted by WC$_k$.

Definition 3. Approximation ratio
Let A be a polynomial time approximation algorithm for WHG and $\rho_A : \mathbb{N} \to]0,1]$ be a function. We say that A guarantees approximation ratio ρ_A if:

$$\forall (G,w) \in \mathcal{G}_w, \quad \frac{\lambda_A(G,w)}{\beta(G,w)} \geq \rho_A(n) .$$

WHG is known to be hard to approximate; the following theorem recalls some hardness results for it:

Theorem 1. *If $P \neq NP$, then:*
(i) there exists $\varepsilon \in]0,1[$ such that HG cannot be polynomially approximated with ratio $n^{\varepsilon-1}$ for any nontrivial hereditary property that is false for some clique or independent set. ([8])
(ii) for maximum clique and maximum independent set problems, item (i) holds for every $\varepsilon > 0.5$. ([7])

Without loss of generality, we can assume that an approximation ratio for WHG is at least W/n, where W denotes the sum of the weights and n is the order of the graph instance: in fact the naïve algorithm computing, for every instance (G,w), a vertex of maximum weight (seen as a graph satisfying π) trivially guarantees this ratio.

2.3. An on-line version: LWHG(2)

The on-line version of WHG is denoted by LWHG. We are interested in the case, denoted by LWHG(2), where the instance is revealed in two clusters: at the first step a weighted graph $(G_1 = (V_1, E_1), w_1)$ of order n_1 is revealed and one has to irrevocably decide which vertices of V_1 belong to the solution. Then, the second part of the instance $(G_2 = (V_2, E_2), w_2)$ of order n_2 is revealed together with edges between V_1 and V_2 and one has to complete the solution by vertices of V_2 in such a way that the whole solution satisfies π . G_1 and G_2 are called *clusters*. In our context, we also suppose that the order n and the total weight $W = |w_1| + |w_2|$ of the whole graph are known at the beginning of the on-line process. If Π is a particular case of WHG, we define $L\Pi$ and $L\Pi(2)$ as well (for instance LWS(2), LWK(2) and LWC$_k$(2)). An on-line algorithm LA has to select some vertices of V_1 and V_2 as soon as they are revealed, so that the whole solution satisfies π . The computational complexity of LA is the sum of the complexities of both steps; LA is said to be polynomial if its computational complexity is bounded above by $P(n)$, where P is a polynomial function, and n denotes the order the whole graph. We then denote by (G,w) the whole graph and by $\mathrm{LA}((G,w),G_1)$ the on-line solution computed by LA for the graph (G,w) if G_1 is revealed at the first step. $\lambda_{\mathrm{LA}}((G,w),G_1)$ denotes the value (weight) of $\mathrm{LA}((G,w),G_1)$. For every weighted graph (G_1,w_1) , with $|w_1| = W_1$, for every rational number $W > W_1$ and every integer $n \geq n_1$, let us consider an instance of LWHG(2) for which (G_1,w_1) is revealed at the first step and the whole graph is of size n and of total weight W . We denote by $\mathrm{LA}((G_1,w_1),n,W)$ the set of vertices introduced in the solution by LA at the first step (when G_1 has been revealed) and by $\lambda_{LA}((G_1,w_1),n,W)$ its value.

Definition 4. Competitivity ratio
Let LA be an on-line algorithm for LWHG(2) and $c_{\mathrm{LA}} : \mathbb{N} \to]0,1]$ *be a function. We say that LA guarantees competitivity ratio* c_{LA} *if:*

$$\forall (G = (V,E), w) \in \mathcal{G}_w, \ \ |V| = n, \ \forall V_1 \subset V, \ \frac{\lambda_{\mathrm{LA}}((G,w), G[V_1])}{\beta(G,w)} \geq c_{\mathrm{LA}}(n) .$$

Algorithm 1. LA

begin
  if $w(A(G_1)) \geq w(V_2)\sqrt{\rho(n(G_2))/n(G_2)}$ then
    output $A(G_1)$
  else
    output $A(G_2)$
  fi
end

We denote by PWHG the restriction of WHG to instances for which weights are polynomially bounded: if P is a polynomial function, we restrict ourselves to instances of order n for which the dimension of each weight is bounded above by $P(n)$. Roughly speaking, it corresponds to the case where we allow polynomial-time complexities to be polynomially related to dimensions of weights.

Remark 2. *By multiplying every weight by P(n), one gets an instance of PWHG, that is equivalent, and for which every weight is an integer. So, PWHG reduces to the case of polynomially bounded integer weights.*

3. COMPETITIVE ANALYSIS FOR LWHG(2): ON-LINE REDUCES TO OFF-LINE

The following result is shown in [3]:

Theorem 2. *([3]) LWHG(2) reduces to WHG; the reduction allows us to transform a* $\rho(n)$*-algorithm A for WHG such that* ρ *decreases,* $n\rho$ *increases and,* $\forall (G=(V,E),w) \in \mathcal{G}_w,\ \ \lambda_A(G,w) \geq w(V)/n$, *into an on-line algorithm for LWHG(2) guaranteeing, for every* $\varepsilon > 0$ *and for every instance* (G,w)

$$c_{\mathrm{LA}} \geq \min\left\{\frac{\varepsilon}{(1+\varepsilon)}\rho(n), \frac{1}{(1+\varepsilon)}\sqrt{\frac{\rho(n)}{n}}\right\}.$$

Let us note that the hypothesis "ρ decreases, $n\rho$ increases and, $\forall (G=(V,E),w) \in \mathcal{G}_w,\ \ \lambda_A(G,w) \geq w(V)/n$" is not restrictive. Algorithm 1 (called LA) describes the reduction: it is parameterized by A, an off-line algorithm for WHG that is assumed to guarantee an approximation ratio ρ. LA has the same computational complexity as A. In particular, this reduction preserves polynomial-time complexity, but it also holds if A is not polynomial.

Let us now focus on ratios of the form[1] $\rho(n) = f(n)/n$, where n denotes the order of the graph, f infinitely increases and $f(n)/n$ decreases beyond a value n_0. In [3], we deduce the following corollary:

Corollary 1. *If A is an approximation algorithm for WHG achieving ratio of the above form, then:*
(i) for every $\varepsilon > 0$, *there exists a constant* $K(\varepsilon)$ *such that, for every graph with* $n > K(\varepsilon)$, *LA achieves competitivity ratio*

$$c_{\mathrm{LA}} \geq (1+\varepsilon)^{-1}\sqrt{f(n)}/n.$$

(ii) If furthermore, $n(G_1) = n(G_2) = n/2$,

$$c_{\mathrm{LA}} \geq 2(1+\varepsilon)^{-1}\sqrt{f(n/2)}/n.$$

---

[1] Most of the known approximation results for problems of the class WHG involve such ratios.

First item is immediately deduced from theorem 2 while the second one is deduced from a slight improvement of its proof in the case where $n(G_1) = n(G_2)$.

A polynomial-time algorithm for WHG guaranteeing approximation ratio $O(\log n / n)$ is devised in [6]. For the special case of maximum weighted independent set WS, this ratio can be improved ([6]) to $O(\log^2 n / n)$. We immediately deduce the following corollary (item (ii) corresponds to the case where A is an exact algorithm for WHG):

Corollary 2.
(i) LWHG(2) admits a polynomial-time on-line algorithm guaranteeing competitivity ratio $O(\sqrt{\log n} / n)$.
(ii) for every $\varepsilon > 0$, *there exists a constant* $K(\varepsilon)$ *such that LWHG(2) admits an on-line algorithm guaranteeing, for every graph of order at least* $K(\varepsilon)$, *competitivity ratio* $(1+\varepsilon)^{-1}(1/\sqrt{n})$.
(iii) LWS(2) admits a polynomial-time on-line algorithm guaranteeing competitivity ratio $O(\log n / n)$.

4. HARDNESS RESULTS

In this section, given an approximation algorithm for WHG guaranteeing ρ, we suppose that there exist a weighted graph (G_1, w_1) and a constant $\nu > 0$ such that:

$$\beta(G_1, w_1)\rho(n(G_1)) \leq \lambda_A(G_1, w_1) < (1+\nu)\beta(G_1, w_1)\rho(n(G_1)) . \tag{1}$$

This condition means that the ratio cannot be widely improved for this algorithm. In the case where the ratio is asymptotically reached, ν can be chosen arbitrary small. Then, we devise the following proposition:

Proposition 3. *Let A be an approximation algorithm guaranteeing* ρ *for WHG and satisfying relation 1 for a constant* ν*; then, algorithm 1 parameterized by A cannot guarantee a competitivity ratio strictly better than*

$$(1+\nu)\sqrt{\frac{\rho(n/2)}{n/2}}$$

even if both clusters have the same order.

Proof: Let $\varepsilon \in]0,1[$, and let (G_1, w_1) be an instance of WHG (of order n_1 and of total weight W_1) satisfying relation 1, for a constant ν. Let $n = 2n_1$ and let W_2 be a rational number such that:

$$\frac{1}{1+\varepsilon}\sqrt{\frac{n}{2}\rho\left(\frac{n}{2}\right)}\beta(G_1, w_1) < W_2 \leq \sqrt{\frac{n}{2}\rho\left(\frac{n}{2}\right)}\beta(G_1, w_1) . \tag{2}$$

Let us consider an on-line instance of LWHG of order n and of total weight $W_1 + W_2$ for which (G_1, w_1) is revealed at the first step and the second cluster G_2 is a graph of order $n/2$, of total weight W_2, and satisfying π. Let us consider the solution computed by algorithm 1 for this instance. From relations 1 and 2 we have $(n_2 = n/2)$:

$$w(A(G_1, w_1)) = \lambda_A(G_1, w_1) \geq W_2 \sqrt{\frac{\rho(n_2)}{n_2}}$$

and consequently, algorithm 1 outputs $A(G_1, w_1)$, while $\beta(G, w) \geq W_2$. Relations 1 and 2 imply that, for every $\varepsilon \in]0,1[$, the related competitivity ratio is bounded above by $(1+\varepsilon)(1+\nu)\sqrt{\rho(n/2)/(n/2)}$, which concludes the proof. ♦

This result means that the analysis of algorithm 1, performed in theorem 2, cannot be significantly improved. In particular, let us point out that, in the case of ratios of the form $f(n)/n$ where f increases, the bound becomes

$$2(1+\nu)\frac{\sqrt{f(n/2)}}{n}.$$

Consequently, if the approximation ratio guaranteed by A is asymptotically tight, then item (ii) of corollary 1 is almost tight.

As shown in the next proposition, this result can be extended to a more general class of algorithms including algorithm 1. In what follows, we express each on-line instance as a two-player game: first player reveals the instance while the second one tries to construct the solution. In our context, this game has two steps; at step $i = 1,2,$ player 1 reveals cluster G_i and player 2 decides which new vertices belong to its solution. This way of describing an on-line problem allows devising hardness results in this context. A competitive on-line algorithm can be seen as a strategy for player 2 guaranteeing, for every instance and every way this instance can be revealed, a level of quality for the solution. On the other hand, a hardness result corresponds to a first player's strategy (for revealing instances) forcing the second one to choose a relatively bad solution. In our context, the order n and the total weight W of the instance are fixed at the beginning of the game. Player 1 has to reveal both clusters (G_1, w_1) of order n_1 and (G_2, w_2) of order n_2 with $n_1 + n_2 = n, |w_1| + |w_2| = W$, so that the solution constructed by player 2 cannot exceed the hardness threshold.

Proposition 4. *Let us consider an approximation algorithm A guaranteeing an approximation ratio ρ that satisfies relation (1) for G_1 and ν. We also suppose that A constructs a maximal solution for every graph and that G_1 does not satisfy π.*

(i) If π satisfies the k-boundary-condition, for a constant k, and if LA is an on-line algorithm selecting, at the first step, either $A(G_1)$ or $\emptyset$, then LA cannot guarantee a competitivity ratio strictly better than

$$2\sqrt{k}\sqrt{1+\nu}\sqrt{\frac{\rho(n_1)}{n_1}}$$

even if both clusters have the same order.

(ii) If, furthermore, $k=1$ *(*π *corresponds to "independent set" or "clique"), then the bound*

$$2\sqrt{1+\nu}\sqrt{\frac{\rho(n_1)}{n_1}}$$

holds even if LA is only supposed to select, at the first step, a subgraph (eventually empty) of $A(G_1)$.

Note that the hypothesis that A constructs a maximal solution and that G_1 does not satisfy π is not restrictive. In fact, every approximation algorithm for WHG can be assumed to devise a maximal (not extendible) solution. In this case, if G_1 satisfies π, then $\mathrm{A}(G_1)=V_1$; but if relation (1) only holds for graphs satisfying π, then the ratio can be easily improved by a multiplicative factor $(1+\nu)$.

Proof:

(i) Let l be an integer (its value will be fixed below), $\varepsilon=1/l$ and $n=n_1/\varepsilon$, where $n_1=n(G_1)$. Let us then choose W_2, a rational number such that:

$$(1+\nu)\beta(G_1,w_1)\rho(n_1)<W_2\sqrt{k}\sqrt{\frac{1+\nu}{\varepsilon(1-\varepsilon)}}\sqrt{\frac{\rho(n_1)}{n_1}}\leq\frac{(1+\nu)}{\varepsilon^2}\beta(G_1,w_1)\rho(n_1). \quad (3)$$

Second player's strategy is assumed to be such that he selects, at the first step, either $\mathrm{A}(G_1)$, or no vertex. Let us then suppose that the whole graph is of order n and of total weight $w(G_1)+W_2$. Let us also assume that first player reveals graph G_1 at the first step. At the second step, player 1 has to reveal a graph G_2 of order $n_2=(1-\varepsilon)n$ and of weight W_2. We also assume that all vertices of G_2 have the same weight W_2/n_2.

We then consider two cases according as player 2 selects some vertices of G_1 or not.

Case 1: Some vertices of $\mathrm{A}(G_1)$ are selected.

According to Proposition 2, condition **C3** holds since $\mathrm{A}(G_1)$ is maximal and is not equal to G_1. Then, player 1 reveals a graph G_2 defined by condition **C3** for $V_1'=\mathrm{A}(G_1)$. Player 2 cannot select any vertex at the second step of the game. Consequently, the weight of the on-line solution is bounded above by $\lambda_{\mathrm{A}}(G_1)$, while the optimal value is at least W_2. Using relations (1) and (3) we deduce that the related competitivity ratio satisfies:

$$c_{\mathrm{LA}}\leq\sqrt{\frac{k(1+\nu)}{\varepsilon(1-\varepsilon)}}\sqrt{\frac{\rho(n_1)}{n_1}}. \quad (4)$$

Case 2: no vertex of G_1 is selected.
In this case, player 1 reveals a graph G_2 defined by condition **C2**$_k$; it implies that the on-line solution contains at most k vertices of weight W_2/n_2, while the optimal value is at least $\beta(G_1,w_1)$. By using the fact that $n_1 = \varepsilon n$ and relation (3), we deduce that relation (4) also holds. It concludes the proof of (i).
(ii) If $k = 1$, conditions **C1** and **C2** hold and the proof is the same.
Finally, we choose $l = 2$ (so $\varepsilon = 1/2$) that minimizes the expression $1/\sqrt{\varepsilon(1-\varepsilon)}$, which concludes the proof. ♦

Proposition 3 means that, for almost every WHG-approximation algorithm A, the competitivity analysis of algorithm 1, devised in theorem 2 cannot be significantly improved. Proposition 4 means, in a way, that the related reduction from LWHG(2) to WHG dominates every such reduction selecting, at the first step, either $\mathrm{A}(G_1)$ or $\emptyset$. In particular, it could not be improved by using another threshold. Finally, the following result points out that, for a class of hereditary properties, this reduction is almost optimal.

Theorem 3. *Let us suppose that π satisfies a k-star-boundary condition, then for every $\varepsilon > 0$, an on-line algorithm LA for LWHG(2) cannot guarantee competitivity ratio*

$$(1+\varepsilon)\frac{\sqrt{k}}{\sqrt{n}}$$

even if weights can take only two values.

Proof: Let $\varepsilon \in]0,1[$, let $n_1 \in \mathbb{N}$ be such that:

$$n_1 > k\varepsilon(1+\varepsilon) \text{ and } \sqrt{\frac{k}{n_1}} + \sqrt{4\varepsilon + \frac{k}{n_1}} \leq 2\sqrt{\varepsilon}\sqrt{1+\varepsilon}. \tag{5}$$

Let also $n = n_1(1+1/\varepsilon)$ $(n \in N)$ and let W be a rational number.
We then define

$$r = \frac{1}{2}\left[\frac{k}{n_1} + \sqrt{\frac{k^2}{n_1^2} + \frac{4\varepsilon k}{n_1}}\right].$$

Let us point out that $r^2 - (k/n_1)r - (k\varepsilon/n_1) = 0$ and that relation (5) implies:

$$r \leq \sqrt{\frac{k}{n_1}}\sqrt{\varepsilon}\sqrt{1+\varepsilon} = (1+\varepsilon)\sqrt{\frac{k}{n}} < 1. \tag{6}$$

Following the same method as previously, player 1 has to reveal a weighted graph of order n, of total weight W and with two possible values for weights.

Player 1 first reveals a set V_1 of order $n_1 = n(\varepsilon/(1+\varepsilon))$ satisfying π (recall Remark (1)) and of total weight $W_1 = W(r/(1+r))$. All vertices of V_1 have the same weight $w_1 = W_1/n_1$.

Player 2 selects a set $V_1' \subset V_1$ of weight W_1'. We then consider two cases according as $|V_1'| \le k-1$ or $|V_1'| \ge k$.

Case 1: $|V_1'| \le k-1$.

At the second step, player 1 reveals a graph G_2 of order $n_2 = n_1/\varepsilon > k$ defined by condition **C2**$_k$. Every new vertex is of weight $(W - W_1)/n_2 = (W_1/n_1)(\varepsilon/r)$. Player 2 can choose at most k vertices during the second step and consequently, the value of the on-line solution is at most

$$\frac{(k-1)W_1}{n_1} + \frac{k\varepsilon W_1}{rn_1}$$

while the optimal value is at least W_1 (recall that G_1 satisfies π). The related competitivity ratio satisfies:

$$c_{\mathrm{LA}} \le \frac{k}{n_1}\left(1+\frac{\varepsilon}{r}\right) = r\ .$$

Case 2: $|V_1'| \ge k$.

In this case, player 1 reveals a graph G_2 of order $n_2 = n_1/\varepsilon$ defined by condition **C1.1**$_k$. Then, player 2 cannot select any vertex during the second step. The value of the on-line solution is at most W_1, while the optimal value is at least $W - W_1$ (G_2 satisfies π). So, the competitivity ratio satisfies:

$$c_{\mathrm{LA}} \le \frac{W_1}{W - W_1} = r\ .$$

In both cases, $c_{\mathrm{LA}} \le r$, which concludes the proof by using relation (6). ♦

This result limits the analysis of every (not only polynomial-time) on-line algorithm. In particular, the competitivity ratio devised in item (ii) of corollary 2 is optimal, up to a constant multiplicative factor, and consequently:

Corollary 3. *The reduction LA cannot be significantly improved.*

It gives us a first answer about the relative hardness of WHG and LWHG(2): the former trivially admits an optimal algorithm, while the best competitivity ratio for the latter is $O(1/\sqrt{n})$. But the question remains open for polynomial-time on-line algorithms; the next section is devoted to this question.

5. OFF-LINE REDUCES TO ON-LINE

In this section, we study how an on-line algorithm can be used in order to solve the off-line version of the problem. Let us first remark that the on-line version LWHG(2) is at least as difficult as the off-line one: every instance of WHG can be seen as an instance of LWHG(2) for which the second cluster is empty. It brings to the fore a trivial reduction preserving the ratio (the competitivity ratio simply becomes an approximation ratio). The aim of this section is to transform an on-line algorithm into an off-line one with an improved ratio.

In what follows, we are interested in polynomially bounded versions PWHG and LPWHG(2) of WHG and LWHG(2), respectively. The following result can be seen, in a way, as an "inverse version" of theorem 2:

Theorem 4. *If π satisfies a k-star-boundary condition for fixed k, then PWHG reduces to LPWHG(2); for every $\varepsilon > 0$, the reduction allows us to transform a competitive $c(n)$-algorithm satisfying, for $n \geq 2k,\ c(n) > \zeta / n$, with $\zeta > 2(k-1)$, into an algorithm approximating PWHG within ratio*

$$\rho(n) \geq \frac{1 - \frac{2(k-1)}{\zeta}}{k(1+\varepsilon)} n(c(2n))^2 .$$

Let us first point out that the condition $c(n) > \zeta / n$ is not restrictive since competitivity ratios produced by Theorem 2 are bounded above by $1/[(1+\varepsilon)n]$. If $k = 1$ (case of independent set or clique), then ζ is only supposed to be positive.

Proof: Since k is a fixed integer, PWHG-instances of order less than k can be solved in constant time; consequently we can restrict ourselves to PWHG-instances of order at least k.

Let LA be a polynomial-time on-line algorithm for LPWHG(2) guaranteeing a competitivity ratio c_{LA} which satisfies, if $n \geq 2k,\ c_{\text{LA}} > \zeta / n$, with $\zeta > 2(k-1)$. We define $\alpha = 1 - [2(k-1)]/\zeta$ (of course $\alpha \in]0,1[$). Let P be a polynomial function and (G_1, w_1) be an instance of PWHG, i.e., a weighted graph of order n_1, of total weight $W_1 = |w_1|$, and such that weights are polynomially bounded by P. In what follows, whenever the expression of P is not known, we replace its value by the maximum dimension of weights in G_1. Let finally $\varepsilon \in]0,1[$, and let Q be the polynomial function $Q = \lceil P / \varepsilon \rceil$. We recall that $\text{LA}((G_1, w_1), n, W)$ denotes the set of vertices computed by LA at the first step of the on-line process if (G_1, w_1) is revealed at this step, and the whole graph is of order n and of total weight W.

One can polynomially compute the quantity:

$$\tilde{W}(G_1, w_1) = \frac{1}{Q(n_1)} \underset{l \in \left\{0, \ldots, \left\lceil \frac{2Q(n_1) n_1 W_1}{\zeta - 2(k-1)} \right\rceil \right\}}{\arg\max} \left[\lambda_{\text{LA}} \left((G_1, w_1), 2n_1, W_1 + \frac{l}{Q(n_1)} \right) \right].$$

The related complexity is $\left(1+\left\lceil \frac{2Q(n_1)n_1W_1}{\zeta-2(k-1)} \right\rceil\right) T(2n_1)$ where T denotes the computational complexity of LA. We then consider the following algorithm A for graphs of order at least k:

$$A:(G,w)\mapsto \mathrm{LA}((G,w),2n,|w|+\tilde{W}(G,w))$$

where n denotes the order of G. We denote by $\mathrm{A}(G,w)$ the solution computed by A for instance (G,w), and by $\lambda_{\mathrm{A}}(G,w)$ (or only λ_A if no ambiguity arises) its value. A is a polynomial-time approximation algorithm for PWHG for instances of order at least k. Moreover, let us point out that $\mathrm{LA}((G_1,w_1),2n_1,W_1)\neq\emptyset$ if $W_1>0$ and, consequently, $\lambda_{\mathrm{A}}(G,w)>0$ if $W>0$.

Let (G_1,w_1) be an instance of PWHG of total weight $W_1>0$ and of order $n_1\geq k$, we define

$$\rho_1=\frac{\lambda_A(G_1,w_1)}{\beta(G_1,w_1)}>0$$

and

$$W_2=\frac{1}{Q(n_1)}\left\lceil \frac{Q(n_1)\lambda_{\mathrm{A}}(G_1,w_1)}{\alpha c(2n_1)} \right\rceil.$$

Then, we have (recall that $\alpha c(2n_1)\leq 1$):

$$\lambda_{\mathrm{A}}(G_1,w_1)<\alpha W_2 c(2n_1)\leq \lambda_{\mathrm{A}}(G_1,w_1)+\frac{1}{Q(n_1)}\leq \lambda_{\mathrm{A}}(G_1,w_1)(1+\varepsilon) \tag{7}$$

where the last inequality holds because $\lambda_{\mathrm{A}}(G_1,w_1)\geq 1/P(n_1)$.

Let us then consider an on-line instance of LPWHG(2) where (G,w_1) is revealed at the first step, the whole graph is of order $2n_1$ and of total weight W_1+W_2, and every weight of G_2 is W_2/n_1. Note that weights of this instance are of dimension bounded above by $O(n_1(P(2n_1))^3)$, and that $W_2=\frac{l}{Q(n_1)}$, with $l\leq\left\lceil \frac{2Q(n_1)n_1W_1}{\zeta-2(k-1)} \right\rceil$. In fact, $\lambda_{\mathrm{A}}(G_1,w_1)\leq W_1$ and $\alpha c(2n_1)\geq(\zeta-2(k-1))/(2n_1)$ (recall $2n_1\geq 2k$). Consequently, by definition of $\tilde{W}(G_1,w_1)$ we have:

$$\lambda_{\mathrm{LA}}((G_1,w_1),2n_1,W_1+W_2)\leq \lambda_{\mathrm{A}}(G_1,w_1)<\alpha W_2 c(2n_1). \tag{8}$$

Let us suppose that $\lambda_{\mathrm{LA}}((G_1,w_1),2n_1,W_1+W_2)>0$, and that G_2, revealed at the second step, is defined by condition $\mathbf{C1.2}_k$ recall that $n(G_2)=n_1\geq k$. Then, at most $k-1$ new vertices (of weight W_2/n_1) will be introduced at the second step and $\beta(G,w)\geq\beta(G_2,w_2)=W_2$, where w_2 and w denote the weight system of G_2 and G, respectively. Then (recall that $2n_1\geq 2k$ and $c(2n_1)>\zeta/(2n_1)$):

$$\frac{\lambda_{\mathrm{LA}}((G,w),G_1)}{\beta(G,w)} \le \frac{1}{W_2}\lambda_{\mathrm{LA}}((G_1,w_1),2n_1,W_1+W_2)+\frac{k-1}{n_1}$$
$$< \alpha c(2n_1)+\frac{2c(2n_1)(k-1)}{\zeta}$$
$$= c(2n_1)$$

which contradicts the fact that the on-line algorithm guarantees competitivity ratio c. We deduce that $\mathrm{LA}((G_1,w_1),2n_1,W_1+W_2)=0$. Let us then suppose that the graph G_2, revealed at the second step, is defined by condition $\mathbf{C2}_k$; $n(G_2)=n_1 \ge k$ then the on-line solution will contain at most k vertices of weight W_2/n_1, while $\beta(G) \ge \beta(G_1)$. Since LA guarantees competitivity ratio c, we have:

$$c(2n_1) \le kW_2/(n_1\beta(G_1,w_1)).$$

By using relation 7, we deduce:

$$c(2n_1) \le \frac{k(1+\varepsilon)\lambda_{\mathrm{A}}(G_1,w_1)}{\alpha \mathrm{n}_1 c(2n_1)\beta(G_1,w_1)}$$

which implies that

$$\frac{\lambda_A(G_1,w_1)}{\beta(G_1,w_1)} \ge \frac{\alpha}{k(1+\varepsilon)} n_1[c(2n_1)]^2.$$

This relation being valid for every instance (G_1,w_1) of PWHG, the proof is complete. ♦

It is well-known (see for instance [2]) that, for a large class of problems including WHG, the weighted version reduces to the polynomially bounded version, up to a multiplicative factor $(1-\varepsilon)$, by a simple scaling and rounding process. The combination of both reductions allows us to prove that WHG reduces to LWHG(2).

Theorem 4 allows us to devise hardness results dealing with polynomial-time on-line algorithms; in particular we deduce from theorems 1 and 4 the following corollary:

Corollary 4.
(i) If ε is such that WHG is not polynomially approximated within ratio $n^{\varepsilon-1}$, then a polynomial time on-line algorithm cannot guarantee competitivity ratio $O(n^{\varepsilon/2-1})$.
(ii) If $P \ne NP$, no polynomial-time on-line algorithm for LWS(2) or LWK(2) guarantee competitivity ratio $n^{\varepsilon-1}$ with $\varepsilon > 0.25$.

Algorithm 2. LA_k

begin

$l \leftarrow \lfloor k/2 \rfloor$;

output $A_l(G_1) \cup A_{k-l}(G_2)$

end

6. MAXIMUM K-COLORABLE INDUCED SUBGRAPH PROBLEM

Hardness results stated in theorems 3 and 4 suppose that π satisfies a k-star-boundary condition for some k. In particular, these results do not apply in the case where π is "k-colorable", for $k \geq 2$. The related problem is denoted by WC_k (maximum weighted induced k-colorable subgraph problem). In this section, we show that, for this problem, the situation is completely different: roughly speaking, the on-line version $\mathrm{LWC}_k(2)$ is not more difficult than the off-line one, for $k \geq 2$. Consequently, $\mathrm{LWC}_k(2)$ appears to be "less difficult" than $\mathrm{L}\Pi(2)$, where Π is based on a hereditary property satisfying a k-star-boundary condition.

Theorem 5. *Suppose that, for every $k \geq 1$, there exists an approximation off-line algorithm A_k guaranteeing an approximation ratio $\rho_k(n)$, for every graph of order n. Then, for every $k \geq 2$, there exists an on-line algorithm LA_k for $\mathrm{LWC}_k(2)$ guaranteeing a competitivity ratio c_{LA_k} such that:*

(i) if k is even, then:

$$c_{\mathrm{LA}}(n) \geq \frac{1}{2}\min\{\rho_{\frac{k}{2}}(n(G_1));\ \rho_{\frac{k}{2}}(n(G_2))\}$$

(ii) if k is odd, then:

$$c_{\mathrm{LA}}(n) \geq \frac{1-\frac{1}{k}}{2}[\min\{\rho_{\frac{k-1}{2}}(n(G_1));\ \rho_{\frac{k+1}{2}}(n(G_2))\}].$$

Moreover, LA_k is polynomial if A_l is polynomial for every fixed l.

Let us point out that algorithm LA_k needs algorithms A_l for $l \leq k$. In order to express this result as a reduction, one can consider a generic problem $\tilde{WC}_k$ for which k is included in the instance. Then, the on-line version of this problem reduces to its off-line version.

Proof: For every weighted graph (G,w), we denote by $\beta_k(G,w)$ the optimal value of problem WC_k for instance (G,w). We then consider algorithm 2 (called LA_k) for $\mathrm{LWC}_k(2)$. It constructs a feasible solution since $\mathrm{A}_l(G_1)$ is l-colorable and $\mathrm{A}_{k-l}(G_2)$ is $(k-l)$-colorable. On the other hand, LA_k is clearly polynomial if A_h is polynomial for every fixed h.

Let us now analyze the competitivity ratio of LA_k. We first point out that $\beta_k(G,w) \leq \beta_k(G_1,w_1) + \beta_k(G_2,w_2)$. On the other hand, if $l \leq k$, then the l heaviest color-classes of a k-colorable subgraph G_k constitute a l-colorable subgraph of G, denoted by G_l. Moreover, the average weight of color classes of G_l is not less than the average weight of color classes of G_k. Consequently, we have:

$$\forall(G,w)\in\mathcal{G}_w,\quad \beta_k(G,w)\le\frac{k}{l}\beta_l(G,w)\,. \tag{9}$$

(i) Let us first suppose that k is even, which implies that $l=k-l=k/2$. Then by using relation (9), we have $\beta_k(G,w)\le 2\beta_l(G,w)$, and consequently:

$$\begin{aligned}\lambda_{\mathrm{LA}_k}(G,w)&=\lambda_{\mathrm{A}_l}(G_1,w_1)+\lambda_{\mathrm{A}_l}(G_2,w_2)\\&\ge\frac{1}{2}\min\{\rho_l(n_1);\ \rho_l(n_2)\}(\beta_k(G_1,w_1)+\beta_k(G_2,w_2))\\&\ge\frac{1}{2}\min\{\rho_l(n_1);\ \rho_l(n_2)\}(\beta_k(G,w)).\end{aligned}$$

It concludes the proof of item (i).
(ii) By using relation (9), we have $\beta_k(G,w)\le(2k/(k-1))\beta_l(G,w)$, and $\beta_k(G,w)\le$ $\le(2k/(k+1))\beta_{k-l}(G,w)$. Then, the proof is the same as for case (i). ♦

In [4], we have reduced WC_k to WS:

Proposition 5. *([4]) For every* $k\ge 2$, WC_k *polynomially reduces to* WS*; the reduction transforms a polynomial-time algorithm* $\rho_{\mathrm{WS}}(n)$ *(n being the order of the instance) into a polynomial-time algorithm for* WC_k *guaranteeing:*

$$\rho_{\mathrm{WC}_k}(n)\ge\rho_{\mathrm{WS}}(kn)\,.$$

For the on-line version of WC_k, we deduce:

Corollary 5. *Suppose that WS can be (polynomially) approximated within* $\rho_{\mathrm{WS}}(n)$*, a decreasing function. Then,* $\mathrm{LWC}_k(2)$ *admits a (polynomial-time) on-line algorithm guaranteeing competitivity ratio* $c_{\mathrm{LWC}_k(2)}$ *such that:*
(i) if k *is even, then:*

$$c_{\mathrm{LWC}_k(2)}(n)\ge\frac{1}{2}\rho_{\mathrm{WS}}\left(\frac{1}{2}\tilde{n}\right)$$

(ii) if k *is odd, then:*

$$c_{\mathrm{LWC}_k(2)}(n)\ge\frac{1-\dfrac{1}{k}}{2}\rho_{\mathrm{WS}}\left(\frac{k-1}{2}\tilde{n}\right)$$

where $\tilde{n}=\max(n(G_1),n(G_2))$.

Finally, by using the $O(\log^2 n/n)$ approximation algorithm for WS ([6]), we get:

Corollary 6. $\mathrm{LWC}_k(2)$ *admits a polynomial-time on-line algorithm guaranteeing competitivity ratio* $O(\log^2 n/n)$.

Considering theorem 5, let us point out that an optimal algorithm for WC_k brings to the fore an on-line algorithm for $\mathrm{LWC}_k(2)$ guaranteeing competitivity ratio 1/2, if k is even, and $(1-1/k)/2$, if k is odd. The following proposition shows that this result is optimal, which means that the reduction devised in theorem 5 cannot be improved.

Theorem 6. *Let* $k \geq 1$.
(i) If k is even, then no on-line algorithm for $\mathrm{LWC}_k(2)$ *can guarantee a competitivity ratio strictly better than* $1/2$.
(ii) If k is odd, then no on-line algorithm for $\mathrm{LWC}_k(2)$ *can guarantee a competitivity ratio strictly better than* $(1-1/k)/2$.

Proof: Let $\alpha \geq 0$ be an integer, let us consider an on-line instance (G,w) of $\mathrm{LWC}_k(2)$, of total order $k\alpha + (k\alpha)^2$, and for which every weight is equal to 1. At the first step, player 1 reveals (G_1,w_1), where G_1 is a balanced complete k-partite graph of order $n_1 = k\alpha$: $G_1 = (V_1^1 \cup \cdots \cup V_1^k, E_1)$, where $|V_1^i| = \alpha$, and two vertices of G_1 are linked by an edge in E if and only if they do not belong to the same set V_1^i. Every weight of w_1 is equal to 1. The second player selects a subgraph G_1' of G_1. We denote by l the chromatic number of G_1', of course $l \leq k$. Let us then consider two cases:

Case 1: $l < k/2$.
Since the independence number of G_1 is α, we have $w(G_1') < (k/2)\alpha$, while $\beta_k(G_1,w_1) = n_1 = k\alpha$. In this case, we suppose that G_2, revealed at the second step, is a clique of order $n_2 = n_1^2$, V_1 and V_2 being linked by a complete bipartite graph and every weight of V_2 being equal to 1. Then, player 2 selects at most $(k-l)$ vertices of V_2, while $\beta_k(G,w) \geq \beta_k(G_1,w_1) = n_1$. Consequently, in this case, the competitivity ratio is bounded above by:

$$\frac{1}{k\alpha}(l\alpha + (k-l)) \leq \frac{l}{k} + \frac{1}{\alpha}. \tag{10}$$

Case 2: $l \geq k/2$.

In this case, we suppose that $G_2 = (V_2, E_2)$, revealed at the second step, is a balanced complete k-partite graph of order $n_2 = n_1^2$, each color classes being of size $k\alpha^2$. V_1 and V_2 are linked by a complete bipartite graph and every weight of V_2 is equal to 1.

During the second step, player 2 selects at most $(k-l)k\alpha^2$ vertices of V_2, while $\beta_k(G,w) \geq \beta_k(G_2,w_2) = n_2 = (k\alpha)^2$. Consequently, in this case, the competitivity ratio is bounded above by:

$$\frac{1}{k^2\alpha^2}(l\alpha + (k-l)k\alpha^2) \leq \frac{k-l}{k} + \frac{1}{\alpha}. \tag{11}$$

(i) If k is even, then relations 10 and 11 imply that the competitivity ratio is bounded above by $1/2+1/\alpha$.
(ii) If k is odd, then $l=(k-1)/2$ and relations 10, and 11 imply that, in both cases, the competitivity ratio is bounded above by $(1-1/k)/2+1/\alpha$.

We complete the proof by choosing α as large as needed.

7. CONCLUSION

In this paper, we have studied some links between the approximation behaviour of a combinatorial problem, and the competitivity behaviour of its on-line version. The considered problem is WHG; it admits numerous well-known particular cases. The on-line version we focus on corresponds to the case where the instance is revealed in two clusters. It can be seen as the *easiest on-line version of WHG*. So, it is suitable for understanding the extra difficulty of the on-line case with respect to the off-line one.

We have first considered hereditary properties satisfying a k-star boundary condition. Independent set, clique, of fixed maximum degree ... satisfy this condition, while k-colorable does not. For this case, we have pointed out a hardness gap between off-line and on-line frameworks: if $\rho(n)$ denotes the approximation threshold for the off-line problem then $\sqrt{\rho(n)/n}$ is, up to a constant multiplicative factor, the competitivity threshold of its on-line version. In order to show that, we have established a reduction (from off-line to on-line) allowing us to transform an on-line algorithm into an off-line one, with improved ratio. It can be seen as the "inverse reduction" of a reduction from on-line to off-line performed in [3]. This reduction being polynomial, it allows us to deduce hardness results for polynomially computable on-line algorithms, from hardness results known in the framework of polynomial approximation. To our opinion, it would be interesting to devise such reductions, from off-line to on-line, for other combinatorial problems.

For the case of maximum k-colorable induced subgraph problem, we have pointed out a completely different behaviour: the on-line problem appears to be, up to a constant multiplicative factor, as efficiently solvable as the off-line one. This theorem improves a result of [3]; it can simply be extended to the case of p clusters, for a fixed p. On the other hand, this result only exploits the fact that, for the considered problem, every feasible solution can be divided into two feasible solutions of close problems. Consequently, our process could be used for other problems satisfying a similar property.

Let us finally point out that the proofs of our hardness results are not valid for the case where competitivity ratios depend on the maximum degree. But numerous approximation ratios for graph problems are expressed with respect to this parameter. So, the problem we are now interested in is to devise hardness competitivity thresholds that are depending on the maximum degree.

REFERENCES

[1] Berge, C., *Graphs and Hypergraphs*, North Holland, Amsterdam, 1973.

[2] Crescenzi, P., Silvestri, R., and Trevisan, L., "On weighted *vs* unweighted versions of combinatorial optimization problems", to appear in Information and Computation.

[3] Demange, M., Paradon, X., and Paschos, V.Th., "On-line maximum order induced hereditary subgraph problems", in: V. Hlavac, K.G. Jeffery, and J. Wiedermann (eds.), *Sofsem 2000 - Theory and Practice of Informatics, volume 1963 of LCNS*, Springer-Verlag, 2001, 326-334.

[4] Demange, M., and Paschos, V.Th., "Towards a general formal framework for polynomial approximation", *Les Cahiers du Lamsade, Vol 177*, Universite Paris-Dauphine, Paris, 2001.

[5] Garey, M.R., and Johnson, D.S., *Computers and Intractability. A Guide to the Theory of NP-completeness*, W. H. Freeman, San Francisco, 1979.

[6] Halldórsson, M.M., "Approximations of weighted independent set and hereditary subset problems", *Journal of Graphs Algorithms and Applications*, 4 (1) (2000) 1-16.

[7] Hástad, J., "Clique is hard to approximate within $n^{1-\varepsilon}$", *Acta Mathematica*, 182 (1999) 105-142.

[8] Lund, C., and Yannakakis, M., "The approximation of maximum subgraph problems", *Proc. 20th Int. Colloquium on Automata, Languages and Programming, volume 700 of LNCS*, Springer-Verlag, 1993, 40-51.