Yugoslav Journal of Operations Research
21 (2010), Number 1, 11-28
DOI: 10.2298/YJOR1101011D

ON-LINE COMPUTATION AND MAXIMUM-WEIGHTED HEREDITARY SUBGRAPH PROBLEMS

Marc DEMANGE

ESSEC Business School, Bucharest, Romania,
demange@essec.edu

Bernard KOUAKOU

Université de Montréal, Montréal, Canada,
bernard.kouakou@gmail.com

Eric SOUTIF

CEDRIC, CNAM, Paris, France,
eric.soutif@cnam.fr

Received: June 2009 / Accepted: March 2011

Abstract: In this paper[1] we study the on-line version of maximum-weighted hereditary subgraph problems. In our on-line model, the final instance (a graph with n vertices) is revealed in t clusters, $2 \leq t \leq n$. We first focus on an on-line version of the maximum-weighted hereditary subgraph problem. Then, we deal with the particular case of the independent set problem. We are interested in two types of results: the competitive ratio guaranteed by the on-line algorithm and hardness results that account for the difficulty of the problems and for the quality of algorithms developed to solve them.

Keywords: On-line algorithms, hereditary property, independent set, competitive ratio.

MSC: 68Q25, 68W40, 90C27, 90C35

1. INTRODUCTION

On-line computation aims to solve combinatorial optimization problems with the constraint that the instance is not a priori completely known before one begins to solve it. In other words, the data set is revealed step-by-step and one has, at each step, to

[1] A preliminary extended abstract appeared in the *Proceedings of 16th International Symposium on Algorithms and Computation, ISAAC 2005.*

irrevocably decide on the final solution dealing with this step. On-line algorithms concern a large class of problems subjected to time constraints (decisions are made before one knows all the data).

In [13], a general framework for on-line problems is drawn. An on-line problem is defined by:

- a combinatorial optimization problem **P**,
- a set of rules R detailing how the final instance is revealed,
- a set of rules R' the algorithm has to respect when making decisions.

The triplet (P,R,R') is also called on-line version of P and (R,R') is called an on-line model. This definition encapsulates usual on-line framework but also the so-called "semi on-line" case for which the instance is revealed in a limited number t of steps ($t > 1$).

1.1. Maximum weighted hereditary subgraph problems

In this paper, we deal with on-line versions of the problem of finding, in a weighted graph G, a maximum-weighted subgraph satisfying a non-trivial hereditary property π.

Let us consider a graph property π assigning to each graph G a value π (G) belonging to $\{0, 1\}$. G is said to satisfy the property π if and only if π $(G) = 1$.

If $G = (V, E)$ is a graph, then a set of vertices $V' \subset V$ is said to satisfy π if and only if the induced subgraph $G[V']$ satisfies π.

Definition 1. *A graph property π is hereditary if, whenever it is satisfied by a graph, it is also satisfied by every induced subgraph; it is non-trivial if it is satisfied by an infinite number of graphs and unsatisfied by an infinite number of graphs.*

Let us also point out that, since π is assumed to be non trivial, then for every n there exists a graph of order at least n satisfying π; by heredity it immediately follows that for every size $n \geq 1$ there exists a graph of order n satisfying π. In particular an isolated vertex trivially satisfies π.

Definition 2. *Given a graph G = (V, E), the maximum hereditary subgraph problem, HG, consists of finding a subgraph of G of maximum order satisfying a given non-trivial hereditary property π.*

WHG denotes the weighted version of *HG*: the input is a weighted graph (each vertex x has a weight *w(x))* and one looks for a maximum-weighted subgraph satisfying *π*.

More precisely, *HG* denotes a class of problems, one for each property. Nevertheless, when no ambiguity occurs, every reference to *(W)HG* corresponds to a specific property.

This class contains natural combinatorial problems such as the maximum independent set problem, the maximum clique problem and the maximum *k*-colorable subgraph problem.

Some on-line versions of *HG* have been studied in [6]. The weighted version, *WHG*, of *HG* has also been handled in the on-line framework (see in [5]). However the method used only works for the case where the instance is revealed in two steps.

This paper is devoted to extend the results to the case where the instance is revealed in t steps, with $t \geq 2$.

1.2. On-line models and competitive analysis

In this paper, we deal with the following on-line version of *WHG*, denoted by *LWHG(t)*, where $t \leq n$:

R: the graph G is revealed in t steps. At each step, a weighted subgraph $G_i = (V_i, E_i)$ (called cluster) of G is revealed together with the edges between vertices of G_i and already revealed vertices. Consequently, at step i the already revealed graph is $G[V_1 \cup ... \cup V_i]$ the subgraph of the whole instance induced by $[V_1 \cup ... \cup V_i]$. The total number of vertices and the total weight are supposed to be known in advance.

R': at each step, one irrevocably decides which new vertices are introduced in the solution.

The hypothesis that the total number of vertices and the total weight are known in advance is not restrictive. Indeed, supposing these quantities unknown, it is easy to see that no interesting result can be found: only the trivial ratio $\frac{W_{\min}}{W(G) - W_{\min}}$ can be guaranteed, where $W(G)$ and W_{min} denote respectively the total and the minimum weight of the final graph G (see below for a precise definition of the competitive ratio).

We also note that in the notation *LWHG(t)*, t can be either a constant or a function on n. We will essentially consider "semi-on line" cases for which t is a constant or eventually $t << n$.

In fact, it is pointed out in [6] that even in the unweighted case, if vertices are revealed one by one ($t = n$), then no significant positive result can be stated. If P denotes a sub-problem of *WHG* - for instance the unweighted case (*HG*) or the maximum (weighted) independent set problem *(W)IS* - we also denote by *LP(t)* the related on-line version.

Such a model can be equivalently seen as a two players game ruled by R and R'. One player, the adversary, generates on-line requests according to rules R. The other player is the algorithm; it responds to requests by taking decisions that allow constructing the solution over the time. For a maximization problem, the adversary's goal is to minimize the ratio of the algorithm's value to the optimal value, while the on-line algorithm's goal is to maximize it.

In the on-line framework, the so-called competitive ratio is a popular way for analyzing the quality of an algorithm.

Let us consider a maximization on-line problem P, an instance σ of P and an algorithm LA for solving P. Furthermore, we consider a function $c_{LA}(n)$. Let $LA(\sigma)$ denote the value of the solution of P computed by the algorithm LA and $\beta(\sigma)$ the on-line optimal value. More precisely $\beta(\sigma)$ can be seen as the best value an algorithm can return if the way the graph is revealed is known in advance.

In most cases, and especially for *LWHG(t)*, the on-line optimal value and the off-line optimal value are equal.

The algorithm LA is said to guarantee the competitive ratio $c_{LA}(n)$ if, for every instance σ of P we have:
$c_{LA}(n) \times \beta(\sigma) \leq LA(\sigma)$ $(c_{LA}(n) \times LA(\sigma) \leq \beta(\sigma)$ for a minimization problem).
LA is also said to be $c_{LA}(n)$-competitive.

Approximation ratio

The competitive ratio looks like the approximation ratio used in polynomial-time approximation theory. This allows considering on-line computation in the continuity of approximation. In the case of approximation, given a polynomial-time algorithm **A** computing, for every instance σ of an NP-hard (off-line) maximization problem a feasible solution, it is said to guarantee an approximation ratio $\rho_A(n)$ if, for every σ instance of order *n*, the approximated value $A(\sigma)$ satisfies:

$\rho_A(n) \times \beta(\sigma) \leq A(\sigma)$ $(\rho_A(n) \times A(\sigma) \leq \beta(\sigma)$ for a minimization problem), where $\beta(\sigma)$ denotes the optimal value of the instance.

For LWHG(t) (with $t << n$), we are interested in links between off-line and on-line algorithms. Indeed, whenever a sub-instance is revealed, one can use an (off-line) approximation algorithm to solve this current sub-instance. A central question in this work is how it is possible to exploit (polynomial-time) algorithms in order to devise (polynomial-time) on-line algorithms and moreover, is it possible to transfer performance guarantees from off-line to on-line framework?

We now recall that both *HG* and *WHG* can be (see [8]) polynomially approximated within $O(\log n / n)$ and this ratio is improved to $O(\log^2 n / n)$ for the maximum and weighted-maximum independent set problems ([4,11]). All these ratios are of the form $f(n)/n$, where is an increasing function with n and $f(n)/n$ decreases for $n \geq \kappa$ where κ is a fixed number). Moreover, every algorithm can easily return a solution which value is at least the average $\frac{w(G)}{n}$ of the weights system. All of this motivates the following hypotheses H, used in the sequel:

Hypotheses H:
Let FWA be a $\rho(n)$-approximation algorithm for *WHG*. We will suppose that:

1. FWA guarantees a ratio of the form $\rho(n) = \frac{f(n)}{n}$, where f is an increasing function with n, and ρ decreases for $n \geq \kappa$, with $\kappa \geq 0$.
2. Algorithm FWA satisfies $w(FWA(G)) \geq \frac{w(G)}{n}$, $\forall G$ (the ratio $1/n$ is always guaranteed, i.e. $f(n) \geq 1$).

Structure of the paper

In section 2, we show that *LWHG(t)* reduces to *WHG*, which allows to derive competitive algorithms from approximation ones. More precisely, if the on-line case admits an approximation ratio of the form $f(n)/n$, then there is an on-line algorithm with a competitive ratio $c_{LWA}(G)$ satisfying:

$$c_{LWA}(G) \geq \left(\frac{1-f(n)^{1/t}}{1-f(n)} \right) \frac{f(n)}{n}$$

For *t = 2*, it leads to a result of [5] which was obtained by adapting to *LWHG(2)* the methodology used for *LHG(t)*. The problem of generalizing theses results to *LWHG(t)*, $t \geq 3$, was raised in this previous paper. The proof of our result is totally different while the on-line algorithm is quite similar. The main idea can be described as follows: at each step, one computes a solution of the problem restricted to the new cluster by using the off-line approximation algorithm (the subroutine) and then one decides either to include the whole solution performed if its value is sufficiently good or to reject it in the opposite case. Such an algorithm is called a *threshold algorithm*.

In section 3 we perform lower bounds for a class of algorithms including threshold ones. The main result is shown in the case where property π is either "clique" or "independent set". Then, we adapt it to a more general subclass of hereditary problems. It points out that, for a large class of *WHG*, our analysis is tight.

Finally, section 4 deals with the on-line independent set problem for which we propose a hardness result that bounds below the competitive ratio of any algorithm (not only threshold ones). It points out that, for this problem, threshold algorithms are almost optimal (among on-line algorithms).

Notations

We will consider only simple undirected graphs *G = (V, E)*, $n = \|V\|$ denotes the order of *G*. Every edge in *E* will be denoted either by *(i, j) = (j, i)* or simply by *ij = ji*. $\overline{G} = (V, \overline{E})$, $\overline{E} = \{(i,j), i \neq j, ij \notin E\}$ denotes the complement of *G*. Given $V' \subset V$, $G[V']$ denotes the subgraph of *G* induced by V'. An independent set of *G* is a set of vertices which are mutually not linked by an edge: $V' \subset V$, $\forall (i,j) \in V' \times V'$, $ij \notin E$; in other words $G[V']$ has no edge. A clique of *G* is a set of vertices V' so that $G[V']$ is a complete graph or, equivalently, V' is an independent set of $\overline{G}$.

2. COMPETITIVE RATIO FOR *LWHG(T)*

This section is devoted to prove the following result:

Theorem 1. *Suppose that WHG admits a polynomial-time* $\rho(n)$ *-approximation algorithm* FWA *with* $\rho(n) = \frac{f(n)}{n}$; *then, under hypotheses H, there exists an on-line polynomial-time algorithm LWA , for LWHG(t), achieving for every graph G of order n a competitive ratio of:*

$$c_{LWA}(G) \geq \frac{1 - f(n)^{1/t}}{1 - f(n)} \rho(n)$$

Moreover, for $\varepsilon > 0$ and n large enough, we have:

$$c_{LWA}(G) \geq (1 - \varepsilon) \frac{f(n)^{1/t}}{(n)}$$

Proof. Let us consider the following algorithm LWA that accepts as input a graph G revealed in t subgraphs (called clusters) $G_1, G_2, ..., G_t$ of order $n_1, n_2, ..., n_t$, respectively, and returns the solution LWA (G) for problem LWHG(t).

It can be seen as a threshold-algorithm with threshold

$$\frac{W - w(G_i)}{r - n_i} f(r - n_i)^{1/t}$$

that only depends on the already known part of the instance.

Algorithm LWA

```
1. $i \leftarrow 1;$
2. $W \leftarrow w(G)$
3. $r \leftarrow n;$
4. while $w\,(FWA\,(G_i)) < \frac{W - w(G_i)}{r - n_i} f(r - n_i)^{1/t}$ and $i < t$ do
5.     $W \leftarrow W - w(G_i);$
6.     $r \leftarrow r - n_i;$
7.     $i \leftarrow i + 1;$
8. end while
9. return $LWA(G) \leftarrow FWA\,(G_i)$
```

This algorithm is polynomial in the order *n* of the instance *G* since we use at most *n* times ($t \leq n$) the polynomial algorithm FWA for instances of order lower than *n*.

For $i \leq t - 1$, we set $R_i = G[V_{i+1} \cup ... \cup V_t]$, and $r_i = n_{i+1} + ... + n_t$; the condition of the *while* loop (in the algorithm LWA) becomes: $w\,(FWA\,(G_i)) < \frac{w(R_i)}{r_i} f(r_i)^{1/t}$.

Denote by $k \leq t$ the value of *i* at the end of the *while loop*. We will distinguish two cases, whether the algorithm returns a non empty solution before the last cluster (i.e., $k < t$) or not (i.e. $k = t$).

In the first case *(k < t)*, revisiting the *while* loop in algorithm LWA, the following statements hold:

$$w(FWA\,(G_i)) < \frac{w(R_i)}{r_i} f(r_i)^{1/t} \quad \forall i < k \tag{1}$$

$$w(FWA(G_k)) < \frac{w(R_k)}{r_k} f(r_k)^{1/t} \geq \frac{w(R_k)}{r_k}$$

In the second case *(k = t)*, only relation (1) holds.
We first point out the following result which will be used in the sequel.

Lemma 1.

$$\forall i < k,\ w(\ FWA\ (G_i)) \leq f(n)^{\frac{k-1}{t}}\ (\ FWA\ (G_k))$$

Proof. (of lemma 1) Let us first consider that *k < t.* For every *i < k,* since $w(R_i) = w(G_{i+1}) + ... + w(G_k) + w(R_k)$, relation (1) becomes*:*

$$w(\ FWA\ (G_i)) < \frac{w(G_{i+1}) + ... + w(G_k) + w(R_k)}{n_{i+1} + ... + n_k + r_k} f(r_i)^{1/t}.$$

By using item 2 of hypotheses H we deduce:

$$w(\ FWA\ (G_i))\ \leq \frac{n_{i+1}w(FWA(G_{i+1})) + ... + n_k w(FWA(G_k)) + r_k w(FWA(G_k))}{n_{i+1} + ... + n_k + r_k} f(n)^{1/t}$$

which implies

$$w(FWA(G_i)) \leq f(n)^{1/t} \sup_{i+1 \leq q \leq k} w(FWA(G_q)) \tag{3}$$

Then, by a simple backward induction, we get the result. By setting $i = k - 1$, the inequality (3) initializes the induction process, i.e., $w(\ FWA\ (\ G_k\ )) \leq f(n)^{1/t} w(FWA(G_k))$. Let us then consider the following induction hypothesis: there exists *j*, *1 < j < k* so that:

$$w(FWA(G_i)) \leq f(n)^{\frac{k-i}{t}} w(FWA(G_k)) \forall i = j,...,k-1 \tag{4}$$

By using relation (3), we have:

$$w(FWA(G_{j-1})) \leq f(n)^{1/t} \sup_{j \leq q \leq k} w(FWA(G_q)) \tag{5}$$

On the other hand, since expression (4) holds for all $i = 1,...,k-1$ (and even for *i = k*), and by using relations $f(n) \geq 1$ and $\sup_{j \leq q \leq k} f(n)^{\frac{k-q}{t}} = f(n)^{\frac{k-j}{t}}$ we have:

$$\begin{aligned} \sup_{j \leq q \leq k} w(FWA(G_q)) &\leq \sup_{j \leq q \leq k} (f(n)^{\frac{k-q}{t}} w(FWA(G_k))) \\ &\leq (f(n)^{\frac{k-j}{t}} w(FWA(G_k))) \end{aligned} \tag{6}$$

Then, relation (5) implies:

$$w(FWA(G_{j-1})) \leq f(n)^{\frac{(k-j+1)}{t}} w(FWA(G_k)) \tag{7}$$

The case k = t is similarly solved by putting w(R_t)= 0 and r_t = 0, which concludes the proof of lemma (1). ■

To prove the theorem, consider two cases, whether *k* < *t* or *k* = *t*.

Case 1: algorithm stops with *k* < *t*.
Heredity implies that $\beta(G) \leq \beta(G_1) + ... + \beta(G_k) + \beta(R_k)$.
Moreover, $\beta(R_k) \leq w(R_k) \leq (r_k / f(r_k)^{1/t}) w(FWA(G_k))$, which implies

$$\beta(G) \leq \rho(n_1)^{-1} w(FWA(G_1)) + ... + \rho(n_k)^{-1} w(FWA(G_k)) + \frac{r_k}{f(r_k)^{1/t}} w(FWA(G_k)).$$

Since *f* increases and $\rho(n) = f(n)/n$ decreases (item 1 of hypothesis *H*), we have:

$$\beta(G) \leq \rho(n)^{-1} (w(FWA(G_1)) + ... + w(FWA(G_k)) + f(n)^{\frac{t-1}{t}} w(FWA(G_k)))$$

Then lemma 1 implies

$$\beta(G) \leq \rho(n)^{-1} (f(n)^{\frac{t-2}{t}} + ... + f(n)^{\frac{1}{t}} + 1 + f(n)^{\frac{t-1}{t}}) w(FWA(G_k))$$

and finally

$$\beta(G) \leq \rho(n)^{-1} \frac{1 - f(n)}{1 - f(n)^{\frac{1}{t}}} w(FWA(G_k))$$

The related ratio is:

$$\frac{w(S)}{\beta(G)} \geq \rho(n) \frac{1 - f(n)^{\frac{1}{t}}}{1 - f(n)}$$

Case 2: The algorithm runs until the t^{th} iteration. By using similar arguments as previously, we get:

$$\beta(G) \leq \beta(G_1) + ... + \beta(G_t)$$

$$\beta(G) \leq \rho(n_1)^{-1} w(FWA(G_1)) + ... + \rho(n_t)^{-1} w(FWA(G_t))$$

$$\beta(G) \leq \rho(n)^{-1} (w(FWA(G_1)) + ... + w(FWA(G_t)))$$

$$\beta(G) \leq \rho(n)^{-1} (f(n)^{\frac{t-1}{t}} + ... + f(n)^{\frac{1}{t}} + 1) w(FWA(G_t))$$

So we have:

$$\frac{w(S)}{\beta(G)} \geq \rho(n) \frac{1 - f(n)^{\frac{1}{t}}}{1 - f(n)}$$

Since f is supposed to infinitely increase, the asymptotic competitive ratio of

$(1-\varepsilon)\frac{[f(n)]^{1/t}}{n}$ immediately follows. ■

As the off-line problems *WHG* and *WIS* are respectively approximated within $\frac{\log n}{n}$ and $\frac{\log^2 n}{n}$ we deduce:

Corollary 1. *For fixed values of t and n large enough,*
(i) *LWHG(t) admits a polynomial* $O((\log n)^{1/t} / n)$ *-competitive algorithm;*
(ii) *LWIS(t) admits a polynomial* $O((\log n)^{2/t} / n)$ *-competitive algorithm.*

If we also consider not polynomial-time algorithms, then the result also holds with $\rho(n)=1$. It leads to the following corollary.

Corollary 2. *For fixed values of t, if the threshold algorithm LWA is parameterized by an optimal off-line algorithm (i.e.,* $\rho(n)=1$*) then, LWA guarantees for LWHG(t) (and also for LWIS(t)) the competitive ratio* $O(n^{\frac{1}{t}-1})$.

3. LIMIT OF THRESHOLD ALGORITHMS (HARDNESS RESULT)

In this section, we first devise hardness results limiting the competitiveness of threshold algorithms if the hereditary property is either "clique" or "independent set". Then, we give some results of more general cases. To this purpose, we use some properties **P**, **P'** and **P"** defined in [3]. That will allow us to treat both *clique* and *independent set* problems. **P** is the following property.

P: For any graph $G_1=(V_1,E_1)$, there exists a graph G = (V, E) such that V = $V_1 \cup \{v_2\}, G[V_1]=G_1$ and $\{v_2\}$ is a maximal subset of V satisfying π.

It can be easily shown that **P** holds if and only if π is either clique or independent set. The following properties **P'** and **P"**, used in the sequel, can be immediately deduced from **P**.

P': For any graph $G_1=(V_1,E_1)$ and for any size n_2, there exists a graph G = (V, E) so that $V=V_1 \cup V_2$, $card(V_2)=n_2$, $G[V_1]=G_1$ and every single set $\{x_2\} \subset V_2$ is a maximal subset of V satisfying π.

P" : For any graph $G_1 = (V_1, E_1)$ and for any size n_2, there exists a graph $G = (V, E)$ such that $V = V_1 \cup V_2$, $card(V_2) = n_2$, $G[V_1] = G_1$, V_2 satisfies π and every single set $\{x_2\} \subset V_2$ is a maximal subset of $V_1 \cup \{x_2\}$ satisfying π.

For example, if π is "clique" then, for **P**, we set $G = (V_1 \cup \{v_2\}, E_1)$ (v_2 is an isolated vertex) while for **P'** we have $G = (V_1 \cup V_2, E_1)$, i.e., V_2 is an independent set and each vertex in V_2 has no link with vertices of V_1. As for **P"**, $G = (V_1 \cup V_2, E_1 \cup P_2(V_2))$ where $P_2(X) = \{(x,y) \in X \times X, x \neq y\}$. In other words, V_2 is a clique and each vertex in V_2 is not connected to a vertex in V_1.
If π is "independent set" then, for **P**, $G = (V_1 \cup \{v_2\}, E_1 \cup V_1 \times \{v_2\})$, for **P'**, $G = (V_1 \cup V_2, E_1 \cup P_2(V_2) \cup V_1 \times V_2)$, and for **P"**, $G = (V_1 \cup V_2, E_1 \cup V_1 \times V_2)$.

Let us now consider an approximation algorithm **FWA** for *WHG* satisfying the set of hypotheses **H**. In [3], for the problem *LWHG*(2), a lower bound of $2\sqrt{1+\mu}\frac{\sqrt{f(n)}}{n}$ of the competitive ratio of a threshold algorithm is devised by assuming the following hypothesis $\mathbf{H}'(\mu)$:

there exists G_1, a graph of order n_1 so that:

$\beta(G_1)\rho(n_1) \leq \mathbf{FWA}(G_1) < (1+\mu)\beta(G_1)\rho(n_1).$

If μ is close to 0, then $\mathbf{H}'(\mu)$ only means that the approximation ratio for **FWA** cannot be significantly improved.

We show that this result can be generalized for any $t \geq 2$; moreover, it shows that the analysis performed in the proof of theorem 1 is asymptotically tight.

Proposition 1. *Let us consider that π is either "clique" or "independent set". If **FWA** satisfies **H** and $\mathbf{H}'(\mu)$ for a given approximation ratio $\rho(n) = f(n)/n$ and $\mu > 0$, then the corresponding threshold algorithm (algorithm **LWA**, section 2) cannot guarantee the competitive ratio*

$$t(1+\mu)^{1-\frac{1}{t}}\frac{f(n)^{\frac{1}{t}}}{n}.$$

The result holds even if the sequence of weights is known at the beginning of the on-line process and if clusters are of the same size.
Proof. **FWA** satisfies $\mathbf{H}'(\mu)$. So there exists a graph G_1 of order n_1 such that $\beta(G_1)\rho(n_1) \leq \mathbf{FWA}(G_1) < (1+\mu)\beta(G_1)\rho(n_1)$. We consider *t-1* positive real numbers $x_1, x_2, ..., x_t$ such that:

$$\mathbf{FWA}(G_1) < x_t < ... < x_2 < x_1 < (1+\mu)\beta(G_1)\rho(n_1) \tag{8}$$

Set $w_k = \frac{nx_k}{t}(1+\mu)^{\frac{k-t-1}{t}} f(n)^{\frac{1-k}{t}}, \forall k \geq 2.$

We derive from relation (8):

$$\beta(G_1)\rho(n_1) \le \mathbf{FWA}(G_1) < t(1+\mu)^{\frac{1}{t}} \frac{f(n)^{\frac{t-1}{t}}}{n} w_t \tag{9}$$

$$t(1+\mu)^{\frac{t-k+1}{t}} \frac{f(n)^{\frac{t-1}{t}}}{n} w_k < t(1+\mu)^{\frac{t-k+2}{t}} \frac{f(n)^{\frac{k-2}{t}}}{n} w_{k-1} \quad \forall k, t \ge k \ge 3 \tag{10}$$

$$t(1+\mu)^{\frac{t-1}{t}} \frac{f(n)^{\frac{1}{t}}}{n} w_2 < (1+\mu)\beta(G_1)\rho(n_1) \tag{11}$$

We apply the algorithm **LWA** to a graph of order $n = tn_1$ and of total weight $W = w(G_1) + w_2 + ... + w_t$. We also suppose that G_1 is revealed at the first step.

Then, we apply the following strategy (rules (1) and (2)) for the adversary to reveal a graph with *n* vertices and total weight *W*. Recall that the on-line problem can be seen as a game between two players: the adversary and the algorithm. Here, to derive hardness results, the adversary aims to reveal an instance for which the algorithm performs badly.

rule (1): For $i < t$, if the algorithm selects some vertices in G_i, then the next clusters $G_j, j = i+1,...,t$ to be revealed are obtained by applying $\mathbf{P}''$ to the graph $G'_1 = G[V_1 \cup V_2 \cup ... \cup V_i]$. Moreover, each $G_j, j \ge i+1$, has no connection with the subgraph $G = G[V_1 \cup V_2 \cup ... \cup V_i]$ and each vertex in $G_j, j \ge i+1$, is of weight $\frac{w_j}{n_1}$.
For the *independent set* problem, each $G_j, j \ge i+1$ is an independent set of order n_1; every vertex of G_j is of weight $\frac{w_j}{n_1}$ and is linked to all vertices in $G[V_1 \cup ... \cup V_i]$: (For the *clique* problem, G_j is a clique with no link with $G[V_1 \cup ... \cup V_i]$).

rule (2): For $i < t$, if the algorithm does not select vertices in G_i, then G_{i+1} is a graph of order n_1 satisfying $\mathbf{P}'$ where the graph $G'_1 = G[V_1 \cup V_2 \cup ... \cup V_i]$ plays the role of G_1 in $\mathbf{P}'$. Every vertex in G_{i+1} has a weight equal to $\frac{w_{i+1}}{n_1}$.
For the independent set problem, G_{i+1} is a clique with vertices of weight $\frac{w_{i+1}}{n_1}$ and each vertex in G_{i+1} is linked to all vertices in $G[V_1 \cup V_2 \cup ... \cup V_i]$. (With a similar construction one can deal with the case of the *clique* problem).

We distinguish 3 cases:

<u>*Case 1:*</u>

Suppose that at least one vertex in G_1 has been selected by **LWA**. In this case, the algorithm selects **FWA**(G_1) vertices in G_1 and rule (1) is applied. The construction of the graph prevents to select any other vertex and consequently, **LWA**(G) = **FWA**(G_1). On the other hand, as G_2 satisfies property π, we get: $\beta(G) \geq w_2$.
According to (9) and by successive applications of (10) we get:

$$\frac{\mathbf{FWA}(G_1)}{\beta(G)} < t\frac{w_2}{\beta(G)}(1+\mu)^{1-\frac{1}{t}}\frac{f(n)^{\frac{1}{t}}}{n} \tag{12}$$

$$\frac{\mathbf{LWA}(G)}{\beta(G)} < t(1+\mu)^{1-\frac{1}{t}}\frac{f(n)^{\frac{1}{t}}}{n} \tag{13}$$

<u>*Case 2:*</u>

Suppose that the first selected vertices belong to G_i , for *1 < i < t*. In this case, **LWA**(G) = **FWA**(G_i) since rule 1 is applied to *i*. Moreover, since G_{i+1} satisfies π, we have $\beta(G) \geq w_{i+1}$, and the construction of G_i implies:

LWA(G) = **FWA**(G_i) = $\frac{w_i}{n_1} = \frac{tw_i}{n}$.

Note that relation (8) and hypothesis $\mathbf{H}'(\mu)$ imply that $x_i < (1+\mu)x_{i+1}$, for $i = 1,...t-1$; then, according to (9) and (10) we get:

$$t(1+\mu)^{\frac{t-i+1}{t}}\frac{f(n)^{\frac{i-1}{t}}}{n}w_i < (1+\mu)t(1+\mu)^{\frac{t-i}{t}}\frac{f(n)^{\frac{i}{t}}}{n}w_{i+1}$$

Hence, $n\mathbf{FWA}(G_i) < t(1+\mu)^{1-\frac{1}{t}}f(n)^{\frac{1}{t}}\beta(G)$, and finally:

$$\frac{\mathbf{LWA}(\mathrm{G})}{\beta(G)} < t(1+\mu)^{1-\frac{1}{t}}\frac{f(n)^{\frac{1}{t}}}{n}. \tag{14}$$

<u>*Case 3:*</u>

Let us finally consider the case where the algorithm does not select any vertices in the subgraphs $G_1, G_2, ..., G_{t-1}$. In this case it returns **FWA**(G_t)) which consists of one vertex in G_t, since rule (2) is applied at iteration *t-1*. Therefore, $\mathbf{FWA}(G_t) = \frac{w_t}{n_1} = \frac{tw_t}{n}$, and according to (b) and (c) we have:

$$\beta(G) \geq \beta(G_1) > t(1+\mu)^{\frac{1}{t}}\frac{f(n)^{\frac{t-1}{t}}}{n}w_t\frac{n_1}{(1+\mu)f(n_1)}.$$

As $n = tn_1$ and f is an increasing function, the previous inequality implies that:

$$\beta(G) > \frac{n}{t(1+\mu)^{1-\frac{1}{t}} f(n)^{\frac{1}{t}}} \mathbf{FWA}(G_t)$$

So

$$\frac{\mathbf{LWA}(G)}{\beta(G)} < t(1+\mu)^{1-\frac{1}{t}} \frac{f(n)^{\frac{1}{t}}}{n} \tag{15}$$

Relations (13), (14) and (15) imply that, in every case $c_{\mathbf{LWA}}(n) < t(1+\mu)^{1-\frac{1}{t}} \frac{f(n)^{\frac{1}{t}}}{n}$, which concludes the proof.

Corollary 3. *Under hypotheses* ***H*** *and* $\mathbf{H}'(\mu)$, *the competitive ratio of* ***LWA*** *satisfies for any ε and for n large enough:*

$$(1-\varepsilon)\frac{f(n)^{\frac{1}{t}}}{n} < c_{\mathbf{LWA}}(n) < t(1+\mu)\frac{f(n)^{\frac{1}{t}}}{n}$$

These bounds are tight since the ratio between the upper and the lower bounds is equal to $t\frac{(1+\mu)}{1-\varepsilon}$.

Note that proposition 1 does not use the whole structure of threshold algorithms. It only uses the existence of G_1 and the fact that the on-line algorithm picks at each step at most as many vertices as algorithm **FWA**. In particular, no specific threshold is used in the proof; it leads to the following result.

Corollary 4*. Let us consider that π is either "clique" or "independent set". Under hypotheses* ***H*** *and* $\mathbf{H}'(\mu)$ *for every on-line algorithm* $\mathbf{LA}_{\mathbf{FWA}}$ *based on* ***FWA*** *in the sense that it selects, at each step, either no vertex (an empty sub-solution) or some of vertices of the solution performed by* ***FWA*** *for the current cluster then, for every ε > 0 and for n large enough we have:*

$$c_{\mathbf{LA}_{\mathbf{FWA}}}(n) < t(1+\mu)\frac{f(n)^{\frac{1}{t}}}{n}$$

An extension to more general hereditary properties

The idea of the previous result can be raised in a more general context, obtained by a natural extension of property $\mathbf{P}'$, called the *k*-boundary condition and introduced in [3].

Definition 3*. [3] Let π be a hereditary graph property and let k be an integer. π satisfies the k-boundary condition if, for every* $n \geq k+1$, *there exists a graph G of order n such that no induced subgraph of G of order k + 1 does satisfy* π.

Many hereditary graph properties satisfy this condition for any k (for instance independent set, clique, planar, acyclic, ...). Moreover, it can be simply shown that this condition exactly corresponds to the fact that π is false for some cliques or independent sets. On the other hand, if π is such a property, then it satisfies the following property (see for instance [3]):

$\mathbf{P}'_k$: For any graph $G_1 = (V_1, E_1)$ and any $n_2 \geq k$, there exists a graph $G = (V, E)$ such that $V = V_1 \cup V_2$, $G|V_1| = G_1$, $|V_2| = n_2$ and $\forall V'_2 \subseteq V_2$ with $|V'_2| = k$, and $\forall v \in V \setminus V'_2, V'_2 \cup \{v\}$ does not satisfy π.

To generalize proposition 1, property $\mathbf{P}'_k$ will play the same part as $\mathbf{P}'$. Let us also point out a property P'''' that will replace P''.

Proposition 2. *[3] Let π be a non trivial hereditary property, let $G_1 = (V_1, E_1)$ be a graph and $V'_1 \neq V_1$ be such that V'_1 is a maximal subset of V_1 satisfying π in G_1; then, the following property P'''' holds:*

P'''' : for any $n_2 \geq k$, there exists a graph G = (V, E) such that $V = V_1 \cup V_2, G[V_1] = G_1, \|V_2\| = n_2, V_2$ satisfies π and V'_1 is a maximal subset of V satisfying π.

Proof. (sketch) The idea is very simple: since π is non-trivial there is a graph G_2 of order n_2 satisfying π. Then, pick a vertex $v \in V_1 \setminus V'_1$. Edges between G_1 and G_2 are defined in such a way that every vertex in V_2 has the same neighborhood in V_1 as v. ■

We also suppose (this is not restrictive) that **FWA** computes maximal solutions (for $\subset$). Then, we consider an on-line algorithm selecting, at each step i, either no vertex or **FWA**(G_i).

Let us briefly sketch how to generalize proposition 1. If the on-line algorithm decides to select some vertices at any step, then by assuming that the rest of the graph is revealed using P'''' (rule 1), it cannot select any supplementary vertex. On the other hand, if it selects none, then the next cluster is defined using $\mathbf{P}'_k$ (rule 2) so that it is not possible to select more than k vertices in this cluster.

Finally, in the construction performed in the proof of proposition 1, we replace weight w_i by $w_i / \sqrt{k}$. We get:

Proposition 3. *Let π be a non trivial hereditary graph property satisfying the k-boundary condition. Let **FWA** be an approximation algorithm that computes, for every instance, a maximal set satisfying π and that satisfies **H** and $\mathbf{H}'(\mu)$ for a given approximation ratio $\rho(n) = f(n)/n$ and $\mu > 0$. Then an on-line algorithm selecting at each step i, either no vertex or **FWA**(G_i) cannot guarantee the competitive ratio*

$$t\sqrt{k}(1+\mu)^{1-\frac{1}{t}}\frac{f(n)^{\frac{1}{t}}}{n}$$

4. ON-LINE INDEPENDENT SET PROBLEM: AN HARDNESS RESULT

Theorem 1 holds for maximum independent set problem. The lower bound devised for threshold algorithms also applies to this problem. Roughly speaking, those results show that the only way to improve the competitive ratio of algorithm **LWA** for *LWIS* is to improve the performance guarantee of the off-line algorithm **FWA** used by **LWA**. Nevertheless, this method is limited by the bound $O(n^{(1/t)-1})$ obtained by using an optimal algorithm as **FWA**. So, the following question is raised: is it possible to get a better competitive ratio by using another type of algorithms? In this section, we bring to the fore a negative answer to this question. More precisely, we prove:

***Theorem 2**. Let **LWA** be an on-line algorithm solving LWIS for* $t \geq 2$ *(the graph is revealed in t clusters). Its competitive ratio* $c_{\mathbf{LWA}}$ *satisfies for every n:*

$$c_{\mathbf{LWA}}(n) \leq t\frac{n^{\frac{1}{t}}}{n}.$$

The result holds even if the weights of clusters are known from the beginning of the on-line process.

Proof. Let **LWA** be an on-line algorithm solving LWIS; consider $t \geq 2$ and $n = kt, k \geq 2$.

Set $w_i = k^{1-\frac{i-1}{t}}$ and consider an on-line instance in *t* steps such that the whole graph has *n* vertices, and w_i is the weight of the i^{th} cluster. So, the total weight is $W = k^{\frac{1}{t}}\frac{k-1}{k^{\frac{1}{t}}-1}$.

We apply algorithm **LWA** (an arbitrary on-line algorithm) to a graph of order *n* and of total weight *W*. Each cluster contains *k* vertices. Every weight in the first cluster is equal to 1. Then, we apply the following strategy:

1. If at step $i < t$, **LWA** has not selected any vertex yet, then a clique G_{i+1} of *k* vertices is revealed with all vertices of weight $1/k^{i/t}$ and not linked to already revealed vertices.
2. If **LWA** selects some vertices at step *i*, then clusters $G_{i+1}, G_{i+2}, ..., G_t$ are independent sets of size *k* with vertices of respective weight $\frac{1}{k^{\frac{i}{t}}}, ..., \frac{1}{k^{\frac{t-1}{t}}}$ and (all) linked to an already selected vertex.

We distinguish two cases.

<u>*Case 1*</u>: rule 2 has never been used. In this case the algorithm has only selected one vertex in the last cluster that is a clique; so $w(\mathbf{LWA}(G)) = \frac{1}{k^{\frac{t-1}{t}}}$.

On the other hand $\beta(G) \geq \beta(G_1) = 1$, so

$$\frac{w(\mathbf{LWA}(G))}{\beta(G)} \leq k^{\frac{1-t}{t}} \tag{16}$$

Case 2: rule 2 has been applied at step *i, i < t.* It is clear that only one vertex of weight $\frac{1}{k^{\frac{i-1}{t}}}$ has been selected (in G_i) since only cliques have been revealed until the i^{th} step and all the next vertices are linked to an already selected vertex; it implies: $w(\mathbf{LWA}(G)) = \frac{1}{k^{\frac{i-1}{t}}}$. Moreover, $\beta(G) \geq \beta(G_{i+1}) = \frac{k}{k^{\frac{i}{t}}}$ and consequently:

$$\frac{w(\mathbf{LWA}(G))}{\beta(G)} \leq k^{\frac{1-t}{t}}.$$

In conclusion, relations (16) and (17) imply:

$$c_{\mathbf{LWA}}(n) \leq k^{\frac{1-t}{t}} \leq \left(\frac{n}{t}\right)^{\frac{1-t}{t}}.$$

The resulting values point out that threshold algorithm (with an optimal off-line algorithm as **FWA**) leads almost to the best possible result. Once more, it generalizes the lower bound $(n-1)^{-1/2}$ already known for $t = 2$ [5].

5. CONCLUDING REMARKS

This work points out that known results for *LHG* and *LWHG*(2) can be generalized to the weighted case *LWHG(t).* In approximation theory a usual question concerns the link between weighted and unweighted versions of a given problem (see for instance [4,2]). In particular, *HG* and *WHG* are equivalently solved by polynomial-time algorithms in the off-line case ([4,8]). The situation is rather different in an on-line framework.

Indeed, the comparison between theorem 1 and results of [6] points out a gap between competitive ratios of *LHG(t)* and *LWHG(t)*, asymptotically equal to:

$$\frac{\rho_{\text{LWHG}}}{\rho_{LHG}} \sim f(n)^{\frac{1}{t}-\frac{1}{2}}\sqrt{t}.$$

More generally, weighted problems induce many questions in an on-line framework. In this work, we focused on an on-line model with weights revealed on-line together with vertices. An ordinary question deals with models for which weights and the instance structure are not connected and do not play the same part. In particular, one could either draw a model where the sequence of the weights is revealed at the beginning while the graph is revealed on-line or a model where the graph is known in advance and weights are on-line.

Naturally, the competitive analysis performed in theorem 1 remains valid in these particular cases. In what concerns hardness results, let us revisit the proof of theorem 2 and note that weights are known beforehand. Therefore, this hardness result also holds for a model with weights outside the on-line process.

Let us now consider a dual situation where the graph is fixed and weights are given on-line. The following proposition shows that the same hardness result holds for this model:

Proposition 4. *Let* **LWA** *be an on-line algorithm solving the problem LWIS for* $t \geq 2$ *(the set of weights is completely revealed in t steps); the total weight of each cluster is known in advance. Then, there exists a graph of order n for which the competitive ratio* $c_{\mathbf{LWA}}$ *satisfies:*

$$c_{\mathbf{LWA}}(n) \leq t\frac{n^{\frac{1}{t}}}{n}.$$

Proof. We set: $n = kt, k \geq 2, w_i = k^{1-\frac{i-1}{t}}, \forall i \geq 1, n_i = k$.

So the total weight of the final graph is $W = k^{\frac{1}{t}}\frac{k-1}{k^{\frac{1}{k}}-1}$.

This time, we apply the algorithm **LWA** to a complete graph of order n and of weight W. The first cluster contains *k* weights, each of them equals 1. Then we use the following strategy.

1. If at step $i < t$, **LWA** has not selected any vertex yet, we reveal a set of k identical weights $\frac{w_{i+1}}{k} = \frac{1}{k^{\frac{i}{t}}}$, in order to form the cluster G_{i+1} of total weight $w_{i+1} = k^{1-\frac{i}{t}}$.
2. If **LWA** selects vertices at step *i* then in each of the next clusters, $G_{i+1}, G_{i+2}, \ldots, G_t, k-1$ vertices are of weight 0 and only one vertex concentrates the weight of the whole cluster.

The proof is completed by using the same arguments as in theorem 2.

Let us finally point out the main difference between theorems 1 and 2. The former gives some information not only about general algorithms but also about polynomial-time while the latter does not allow taking into account any considerations about complexity. Theorem 2 points out that threshold algorithms parameterized by an exact off-line algorithm are almost optimal among on-line algorithms. An interesting question is whether the same result holds for polynomial-time on-line algorithms. Theorem 1 induces that any improvement dealing with the approximation of *WHG* would immediately induce an improvement of the competitive ratio that can be guaranteed in polynomial-time. What about the converse? In [3], we propose a reduction from *WIS* to *LWIS(2)* allowing us to show that any improvement of *LWIS(2)*'s competitive ratio guaranteed by any polynomial-time on-line algorithm would imply an improvement of *WIS*'s approximation ratio. A consequence is a hardness result for polynomial-time on-line algorithms. A generalization of this result to *LWIS(t)* or, more generally, the design of such reductions from off-line to on-line seems to be a fruitful research area.

REFERENCES

[1] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., and Protasi, M., *Complexity and Approximation (Combinatorial Optimization Problems and Their Approximability Properties)*, Springer-Verlag, 1999.

[2] Crescenzi, P., Silvestri, R., and Trevisan, L., "To weight or not to weight: where is the question ?", *in: Proc. of 4th Israel Symposium on Theory on Computing and Systems, IEEE Computer Society*, 1996, 68-77.

[3] Demange, M., "Reducing off-line to on-line: an example and its applications", *Yugoslav Journal of Operations Research*, 13(1) (2003) 3-24.

[4] Demange, M., and Paschos, V. Th., "Improved approximations for weighted and unweighted graph problems", *Theory of Computing Systems*, 38(6) (2005) 763-787.

[5] Demange, M., and Paschos, Th.V., "Two-steps combinatorial optimization", in: *Proc. of OLCP'01*, 2001, 37-44.

[6] Demange, M., Paradon, X., and Paschos, Th.V., "On-line maximum-order induced hereditary subgraph problems", *International Transactions in Operational Research*, 12 (2) (2005) 185-201

[7] Garey, M.R., and Johnson, D.S., *Computers and Intractability. A Guide to the Theory of NP-Completness*. CA.Freeman, San Francisco, 1979.

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

[9] Halldórsson, M.M., and Radhakrishnan, J., "Improved approximations of independent set in boundeed-degre graphs via subgraph removal", *Nordic Journal of Computing*, 1 (4) (1994) 475-492.

[10] Halldórsson, M.M., and Radhakrishnan, J., "Greed is good: approximation of independent set in sparse and boundeed-degree graphs", *Algorithmica*, 18 (1) (1997) 145-163.

[11] Halldórsson, M.M., "Approximations via partitioning", JAIST, Japan Advanced, Institute of Science and Technology, Japan, 1995.

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

[13] Paradon, X., "Algorithmique on-line", Ph.D. Thesis, Université Paris Dauphine, 2000.