



HAL
open science

On-line models and algorithms for MAX INDEPENDENT SET

Bruno Escoffier, Vangelis Th. Paschos

► **To cite this version:**

Bruno Escoffier, Vangelis Th. Paschos. On-line models and algorithms for MAX INDEPENDENT SET. pp.16, 2004. hal-00017612

HAL Id: hal-00017612

<https://hal.science/hal-00017612v1>

Submitted on 24 Jan 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On-line models and algorithms for MAX INDEPENDENT SET

Bruno Escoffier*, Vangelis Th. Paschos*

Résumé

Dans un problème on-line, l'instance du problème n'est pas entièrement connue au départ mais est révélée étape par étape, le but étant de construire, au cours de ce processus où l'on découvre l'instance, une solution réalisable la meilleure possible. Nous nous intéressons dans cet article au problème du stable on-line. Les modèles étudiés jusqu'à présent pour le problème du stable on-line consistant, partant d'un graphe vide, à révéler le graphe sommet par sommet, ou par sous ensembles de sommets. Nous allons ici nous intéresser à deux nouveaux modèles du stable on-line. Premièrement, nous étudierons l'approximabilité du problème lorsque le graphe est initialement complet et des arêtes disparaissent à chaque étape, jusqu'à l'obtention du graphe final. Ensuite, nous reprenons les modèles étudiés dans [M. Demange, X. Paradon and V. Th. Paschos, *On-line maximum-order induced hereditary subgraph problems*, Proc. SOFSEM 2000—Theory and Practice of Informatics, LNCS 1963, pp. 326–334, 2000] et proposons une relaxation basée sur l'introduction d'une possibilité de *backtracking* payant.

Mots-clefs : Algorithmique on-line; Ensemble stable, Rapport de compétitivité

Abstract

In on-line computation, instance of the problem dealt is not entirely known from the beginning of the solution process, but it is revealed step-by-step. In this paper we deal with on-line independent set. On-line models studied until now for this problem suppose that the input graph is initially empty and revealed either vertex-by-vertex, or cluster-by-cluster. Here we present a new on-line model quite different to the ones already studied. It assumes that a superset of the final graph is initially present (in our case the complete graph on the order n of the final graph) and edges are progressively

* LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {escoffier,paschos}@lamsade.dauphine.fr

removed until the achievement of the final graph. Next, we revisit model introduced in [M. Demange, X. Paradon and V. Th. Paschos, *On-line maximum-order induced hereditary subgraph problems*, Proc. SOFSEM 2000—Theory and Practice of Informatics, LNCS 1963, pp. 326–334, 2000] and study relaxations assuming that some paying backtracking is allowed.

Key words : Approximation algorithms; Competitive ratio; Maximum independent set; On-line algorithms

1 Introduction

1.1 On-line computation

On-line algorithms have been introduced to tackle situations where problem's solution is planned under uncertainty concerning the final instance of the problem dealt. This kind of situations appears frequently when we have to efficiently solve a problem in real time. In such situations we need to start problem's resolution before the whole instance is completely known. They lead to what is called on-line combinatorial optimization problems. Models for such problems are usually based upon the following two constitutive hypotheses:

1. instance of the problem is revealed step-by-step;
2. decision makers make choices once a part of the instance is revealed, these choices being definite and irrevocable.

Starting from these hypotheses, one can built different models depending on how instance is precisely revealed and what are the rights of the decision maker (on-line algorithm) for constructing the final solution.

Since about twenty years, many combinatorial optimization problems have been studied in on-line versions. For example, [1] studies on-line models for TRAVELLING SALESMAN, [3, 9] study models for on-line MAX INDEPENDENT SET, etc. Also, an interesting survey about on-line combinatorial optimization problems can be found in [10]. In fact, what is easily understood from all these papers, is that on-line-computation is a natural extension of approximation theory.

Given an on-line problem Π , an on-line algorithm for Π provides, for any instance x (and following to the rules of the on-line model describing Π) a feasible solution y for x . The quality of y is measured by the so-called competitive ratio $m(x,y)/\text{opt}(x)$, where $m(x,y)$ is the value of y and $\text{opt}(x)$ the value of the optimal off-line solution for x . We will say that an on-line algorithm A guarantees competitive ratio $f(x)$, if f is a function such that, for any instance x of Π the competitive ratio of solution y computed by A

is better (greater than, or equal to, if we deal with a maximization problem, less than, or equal to, if the problem dealt is a minimization one) than $f(x)$.

1.2 On-line models for MAX INDEPENDENT SET

Given a graph $G(V,E)$, an *independent set* is a subset $V' \subseteq V$ such that whenever $\{v_i, v_j\} \subseteq V'$, $v_i v_j \notin E$, and MAX INDEPENDENT SET consists in finding an independent set of maximum size. In weighted MAX INDEPENDENT SET we consider that vertices are provided with positive weights and the objective becomes to determine an independent set of maximum total weight.

For MAX INDEPENDENT SET, the most natural on-line model seems to be the following one: the initial graph is empty and vertices are revealed one-by-one; together with a “new” vertex all edges linking it with “old” ones are simultaneously revealed. Once a new vertex arrives an algorithm for this model has to decide if this vertex will be included in the solution under construction or not. Such a model is however quite restrictive since no on-line algorithm can guarantee competitive ratio better than $1/(n-1)$, where n is the order of the final graph (while any on-line algorithm trivially achieves this ratio). In [3] a relaxation of this model is proposed. There, instead vertex-by-vertex, G is revealed within $t < n$ clusters. Any time a new cluster arrives all edges linking its vertices with the ones of the older clusters are also revealed. Any on-line algorithm has then to decide which among the vertices of this new cluster have to be integrated in the independent set under construction. It is proved there that, if t clusters are needed that the whole graph is revealed, there exists a polynomial time on-line algorithm achieving competitive ratio $\Omega(\log n / (n\sqrt{t}))$. Two other kinds of relaxations of the basic model are studied in [9]. There, it is assumed that the algorithm can maintain a collection of n^k independent sets (for some constant k) and at the end of the game it can choose the best of the solutions maintained. Under this assumption a competitive ratio $\Omega(\log n/n)$ is achieved. In the second model of [9], the algorithm is allowed to copy intermediate solutions and to extend the copied solutions in different ways. The so-obtained competitive ratio is, once again, $\Omega(\log n/n)$.

Recall that the best known approximation ratios for MAX INDEPENDENT SET and WEIGHTED MAX INDEPENDENT SET are:

- (asymptotical) k/Δ [4] and $3/(\Delta + 2)$ [7], respectively,
- $\Omega(\log^2 n/n)$ [8] (for both versions),
- $\min\{\Omega(\log n/(\Delta \log \log n)), n^{-4/5}\}$ [5] (for both versions also).

In this paper we first study (Section 2) the following on-line MAX INDEPENDENT SET-model: the initial graph is a clique on n vertices and in each step some (one or more) of its edges are removed; any time edges are removed, the on-line algorithm is allowed to add to the independent set under construction vertices adjacent to some of these edges.

Next, in Section 3, we revisit the on-line model already studied in [3] and we relax it by ignoring irrevocability requirement, assuming instead that any time a decision is changed, this change is charged by some non-negative cost. Note that, relaxations dealing with decisions irrevocability also appears in [9], where algorithm is allowed to maintain at each step several solutions in order to finally return the best among them.

In what follows, we denote by n the order of the input-graph G , by Δ its maximum degree and by $\alpha(G)$ the cardinality of a maximum independent set for G (commonly called stability or independence number [2]).

2 On-line edge removal

As we have already mentioned, the model studied in this section consists of starting from a complete graph on n vertices (this is also the order of the final graph) and of supposing that edges disappear step-by-step; at each step, one or more edges are removed. Upon the removal of a set of edges, algorithm has to irrevocably decide which of the vertices adjacent to them are included to the independent set under construction.

We first suppose that the number of steps (iterations) needed that the final graph is fixed is not known in advance to the algorithm (section 2.1). In this case we propose a natural greedy on-line algorithm and show that it is strongly competitive. Next, we suppose that number of iterations needed for fixing the final graph is known in advance. For this case, we devise an on-line algorithm achieving non-trivial competitive ratio in particular when the final graph is fixed within a small number of iterations. For any of the cases dealt we also prove upper bounds to the corresponding competitive ratios.

2.1 The number of iterations is not known in advance

Denote by GA, the natural greedy MAX INDEPENDENT SET algorithm (see, for example, [11] for more details about its approximability). The on-line algorithm considered here is the following, denoted by OLGA:

- at step i , determine the subgraph H_i , induced by the vertices that are adjacent to the edges just removed but non-adjacent to the vertices already included in the independent set S under construction;
- compute $\text{GA}(H_i)$;
- solution at step i becomes $S = S \cup \text{GA}(H_i)$.

Proposition 1 *Competitive ratio achieved by OLGA is bounded below by $2/(n - 1)$, if the parameter dealt for the analysis is the order n of the input graph, or $1/\Delta$, if the parameter dealt is the maximum graph-degree Δ .*

Proof. Assume first that the parameter for the analysis of competitiveness is n . We distinguish two cases, namely, $\alpha(G) = n$ and $\alpha(G) < n$. For the former one, the final graph is simply a set of isolated vertices and there OLGA trivially determines an independent set of cardinality n , achieving so a competitive ratio $1 > 2/(n - 1)$. For the latter case, obviously the final graph contains at least one edge; so, $\alpha(G) \leq n - 1$. Here, OLGA will determine an independent set containing at least the endpoints of a removed edge, i.e., an independent set of cardinality at least 2. The claimed ratio is so proved.

Assume now that the parameter for the analysis of competitiveness is Δ and note that OLGA always computes an independent set maximal for the inclusion. Such an independent set S always guarantees $|S|/\alpha(G) \geq 1/\Delta$ [11]. ■

Theorem 1 *No on-line algorithm can achieve competitive ratio strictly greater than $2/(n - 1)$ or $1/\Delta$ (under the on-line model assumed) for MAX INDEPENDENT SET.*

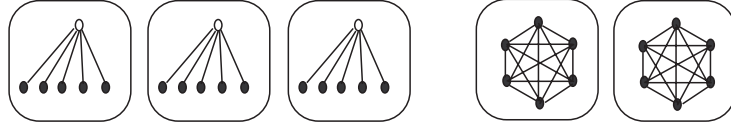
Proof. Denote by A an on-line algorithm for MAX INDEPENDENT SET constructing an independent set S and run it in the following instance:

- during the first iteration, edge (v_1, v_2) is removed;
- if A does not choose any of v_1 or v_2 , game is over;
- otherwise, A has chosen at least one among v_1 and v_2 , say v_1 ;
- then, the second iteration consists of removing all edges in the subgraph of G induced by vertex-set $\{v_2, \dots, v_n\}$.

In the case where A has not made any choice among v_1 and v_2 , we have $|S| = 0$ and $\alpha(G) = 2$. In the case where at least v_1 has been introduced in S , no vertex in $\{v_3, \dots, v_n\}$ can complete it (since all these vertices are linked to v_1). Hence, $|S| \leq 2$, while maximum independent set is $\{v_2, \dots, v_n\}$ of cardinality $n - 1$. In both cases, $|S|/\alpha(G) \leq 2/(n - 1)$, and the proof of the first statement of the theorem.

Fix now a $\Delta > 2$. We will build a graph G of maximum degree Δ and of order $n = k(\Delta + 1)$ (for any $k \geq 2$). Group the n vertices v_1, v_2, \dots, v_n in k groups of $\Delta + 1$ vertices per group (assuming that $v_1, v_2, \dots, v_{\Delta+1}$ are put in the first group, \dots , $v_{(k-1)(\Delta+1)+1}, \dots, v_{k(\Delta+1)}$ are in the k th (last) group).

First iteration consists of removing all edges linking two vertices lying to two distinct groups. We so obtain a non-connected graph on k connected components G_1, G_2, \dots, G_k each of these components been a complete subgraph on $\Delta + 1$ vertices. Here A can choose at most one vertex per graph G_i , $i = 1, \dots, k$ to add it in S . Suppose that it chooses a vertex in each of the graphs G_1, G_2, \dots, G_l , $0 \leq l \leq k$, and no vertex in the rest of the components. In the second and last iteration we remove in G_1, \dots, G_l any edge non-incident to the vertex chosen by A . The form of the graph is as shown in figure 1, assuming $\Delta = 5$, $k = 5$, $l = 3$, as well as that white vertices have been added in S .


 FIG. 1 – $\Delta = 5$, $k = 5$, $l = 3$

In this case, no vertex can be added in S , since all vertices incident to the edges just removed are linked to the vertex added previously; hence, $|S| = l$. On the other hand, there exists an independent set S^* with $|S^*| = l\Delta + (k - l)$ in G containing

- the Δ vertices linked to the vertex chosen by \mathbb{A} in any of G_1, G_2, \dots, G_l and
- one vertex per graph G_{l+1}, \dots, G_k .

So, $|S|/\alpha(G) \leq l/(l\Delta + k - l) \leq 1/\Delta$, which completes the proof of the second statement and of the theorem. ■

Proposition 1 and Theorem 1 immediately conclude the following corollary.

Corollary 1 *Algorithm OLGA is optimally competitive for MAX INDEPENDENT SET (under the model dealt).*

With very similar arguments the following upper bounds can be proved for the case where the final graph is connected.

Theorem 2 *Assuming that the final graph is connected,*

- *no on-line algorithm can achieve competitive ratio better than $2/(n - 2)$, for any graph of order $n \geq 4$;*
- *no on-line algorithm can achieve competitive ratio better than $1/(\Delta - 1)$.*

Furthermore, one can easily prove that OLGA achieves competitive ratios $2/(n - 2)$ and $1/\Delta$ on connected graphs, but not $1/(\Delta - 1)$.

2.2 The number of iterations is known in advance

We assume in this section that number of steps, denoted by t , needed for revealing the graph is known in advance, i.e., it is, in some sense, part of the instance of the on-line MAX INDEPENDENT SET. We also use the following notations: G_i denotes the graph at the end of iteration i ; hence, $G_t = G$ and, since in any iteration we remove some edges, G is a partial subgraph of G_i for any $i \leq t$; I^i denotes the set of vertices adjacent to the set of edges removed during iteration i , i.e., the set of vertices that can be added to the solution under construction in iteration i . Moreover, if V' is a subset of the vertex-set V

of G , $G_i[V']$ denotes the subgraph of G_i induced by V' . For example, $G_t[I^1]$ corresponds to the subgraph of $G_t = G$ induced by the vertices one could add in the solution during the first iteration.

Let us consider, for example, figure 2. The final graph contains 5 vertices $\{1, 2, \dots, 5\}$ and is revealed in three steps. During first iteration, edges $(1, 4)$ and $(2, 3)$ disappear; so, $I^1 = \{1, 2, 3, 4\}$. During second iteration edges $(2, 5)$ and $(3, 4)$ are further removed, and $I^2 = \{2, 3, 4, 5\}$. Finally, during third (last) iteration, edges $(1, 3)$ is also removed; hence, $I^3 = \{1, 3\}$. In figure 3, graphs $G_1[I^1]$, $G_2[I^1]$, and $G_3[I^2]$ are illustrated.

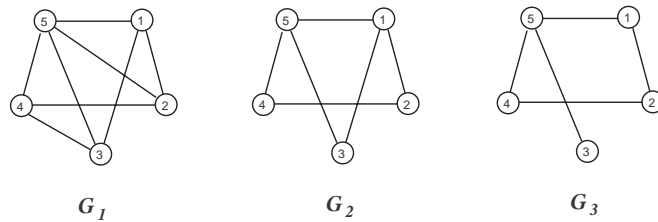


FIG. 2 – The three steps for revealing G .

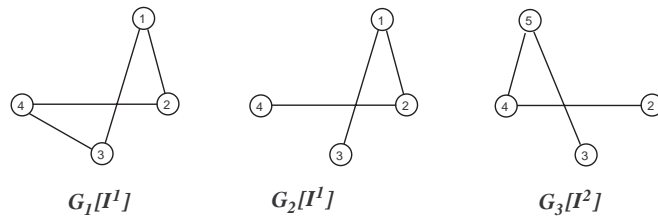


FIG. 3 – Graphs $G_1[I^1]$, $G_2[I^1]$, and $G_3[I^2]$

The following lemma gives an upper bound for $\alpha(G)$ linking it to quantities $\alpha(G_i[I^i])$.

Lemma 1 $\alpha(G) \leq \sum_{i=1}^t \alpha(G_i[I^i])$, and this independently on the way G is revealed.

Proof. Let J^i be the subset of I^i that does not belong to any I^k , $k > i$; J^i corresponds to the vertices that one can choose to put in the solution for last time during iteration i (obviously, J^i can be empty). For example, in figure 2, $J^1 = \emptyset$, $J^2 = \{2, 4, 5\}$ and $J^3 = \{1, 3\}$.

Let S^* be a maximum independent set of G . Remark that vertices of G that do not belong to any J^i are exactly those that are not adjacent to any edge removed; hence they are linked to any other vertex of G and, consequently, they cannot belong to S^* . So, sets $S^* \cap J^i$, $1 \leq i \leq t$ form a partition on S^* (note that some of these sets may be empty).

On the other hand, set $S^* \cap J^i$ is an independent set of G and is included in I^i (because $J^i \subset I^i$). Note also that an independent set of G included in I^i is not mandatorily an independent set of $G_i[I^i]$. In fact, no reason forbids that edges linking two vertices of I^i are removed during an iteration subsequent to iteration i ; for example, in figure 2, set $\{1,3\}$ is an independent set of G but not of $G_1[I^1]$. But under the definition of J^i just above, $S^* \cap J^i$ is indeed an independent set for $G_i[I^i]$. We so obtain: $\alpha(G) = |S^*| = \sum_{i=1}^t |S^* \cap J^i| \leq \sum_{i=1}^t \alpha(G_i[I^i])$, qed. ■

Consider now the following algorithm for on-line MAX INDEPENDENT SET, denoted by OLTA calling as subroutine an approximation MAX INDEPENDENT SET-algorithm AA achieving approximation ratio $\rho(n)$ ($r(n,t)$ is a threshold to be precised later and $\rho(n)$ decreases with n):

- set $i = 1$;
- while $|\text{AA}(G_i[I^i])| < r(n,t)$ and $i < t$, set $i = i + 1$;
- output $S = \text{AA}(G_i[I^i])$.

Theorem 3 *Algorithm OLTA achieves competitive ratio $\sqrt{\rho(n)/(nt)}$ for on-line MAX INDEPENDENT SET (under the model considered). Moreover, it is polynomial if A is so.*

Proof. If OLTA outputs a solution before iteration t , then the size of this solution is $|\text{AA}(G_i[I^i])| \geq r(n,t)$ and the competitive ratio thus achieved is at least

$$\frac{r(n,t)}{\alpha(G)} \geq \frac{r(n,t)}{n} \quad (1)$$

Otherwise, for all $i \in \{1,2,\dots,t\}$,

$$|\text{AA}(G_i[I^i])| < r(n,t) \quad (2)$$

Since AA is assumed to guarantee approximation ratio ρ ,

$$\frac{|\text{AA}(G_i[I^i])|}{\alpha(G_i[I^i])} \geq \rho(|I^i|) \geq \rho(n) \quad (3)$$

Using (2), (3) and Lemma 1, we get:

$$\alpha(G) \leq \sum_{i=1}^t \alpha(G_i[I^i]) \leq \sum_{i=1}^t \frac{|\text{AA}(G_i[I^i])|}{\rho(n)} \leq \sum_{i=1}^t \frac{r(n,t)}{\rho(n)} = \frac{r(n,t)t}{\rho(n)}$$

and consequently,

$$\frac{|\text{OLTA}(G)|}{\alpha(G)} \geq \frac{1}{\alpha(G)} \geq \frac{\rho(n)}{r(n,t)t} \quad (4)$$

Ratio in (1) is increasing with $r(n,t)$, while the one in (4) is decreasing with $r(n,t)$. Equality of them holds for $r(n,t) = \sqrt{\frac{n\rho(n)}{t}}$ and, in this case, ratio achieved is as claimed. ■

Corollary 2 *Dealing with an optimal off-line algorithm AA, competitive ratio implied by Theorem 3 is $1/\sqrt{nt}$. If, on the other hand, AA is the polynomial time approximation algorithm of [8], then the competitive ratio achieved by OLTA is bounded below by $\Omega(\log n/(n\sqrt{t}))$. In particular, when t is fixed, then this ratio is $\Omega(\log n/n)$.*

Theorem 4 *No on-line algorithm can achieve, for the on-line MAX INDEPENDENT SET-model considered, competitive ratio strictly better than $1/(\sqrt{n/2} - 1)$ (for $n \geq 3$), even if $t = 2$.*

Proof. Consider an on-line algorithm A, an integer $n \geq 3$ and set $p = \lfloor \sqrt{2n} \rfloor$ ($p < n$). Assume that first step of graph revealing consists of removing edges in such a way that set $V' = \{v_1, v_2, \dots, v_p\}$ becomes an independent set; assume also that only such edges are removed during this first step. For the second step we distinguish the two following cases:

1. if, during first iteration, A has chosen at least a vertex (vertices chosen belong to V'), then, in the second step, we remove all edges non-incident to any vertex in V' ; in such a case A cannot extend independent set previously constructed while there exists in G an independent set of size $n - p$ composed by all vertices in $V \setminus V'$; so, competitive ratio of A is, in this case, at most $p/n - p$;
2. otherwise, in second step we remove only one edge, say (v_1, v) , where v can be any vertex not in V' ; in this case, the independent set built by A contains at most two vertices (v_1 and v), while set $V' \cup \{v\}$ is an independent set for G of size $p + 1$; consequently, the competitive ratio achieved by A is at most $2/(p + 1)$.

Combination of the ratios of Cases 1 and 2, together with the fact that $p \leq \sqrt{2n} \leq p + 1$, results in a competitive ratio for A bounded above by $1/(\sqrt{n/2} - 1)$. ■

3 Relaxed models and charges

As we have already mentioned, we study in this section two further relaxations of the on-line model introduced in [3]. Recall that in this model:

- graph is revealed by clusters;
- an on-line algorithm builds its solution irrevocably choosing at each iteration which among vertices of the cluster just arrived will be included in the solution under construction.

Relaxations considered for this model are based upon weakening constraints of irrevocability. We will assume that algorithm can, during iteration i , includes in the solution also vertices revealed during iterations $j < i$. Such a relaxation can, of course, be very weak (permissive) since, if no additional assumption is made, algorithm can wait until the whole of graph is revealed before making any choice of the solution; in this case, on-line MAX INDEPENDENT SET-model becomes the classical off-line MAX INDEPENDENT SET. To avoid such situation, we introduce charges penalizing freedom: delayed choice of a vertex will be charged in such a way that its contribution in the final independent set will be smaller than 1; furthermore, the later a vertex chosen, the smaller its contribution in the final solution. More precisely, we will assume that if a vertex revealed during iteration j , is included in the solution during iteration $i \geq j$, then its real value in this solution is $1/k^{i-j}$ (where $k > 1$ is a real number). Under this assumption, one can consider that, in iteration i , algorithm has to run on a vertex-weighted graph, where weights are as follows:

- vertices just arrived (i.e., arrived in iteration i) receive weight 1;
- vertices arrived in iteration $i - 1$ are weighted by $1/k$;
- ...
- vertices arrived in iteration 1 are weighted by $1/k^{i-1}$

Following this model, the real objective for an on-line algorithm A is to compute not really a maximum-size independent set but rather a maximum-weight one. The competitive ratio associated with this model is $\text{val}_A(S')/\alpha(G)$, where S' is the independent set computed by A and $\text{val}_A(S')$ its total weight.

In Section 3.1 we will assume that inclusion of a vertex in the solution under construction is irrevocable. Next, in Section 3.2, we further relax our model assuming that algorithm can backtrack, i.e., that it can even remove from current solution a vertex previously introduced. Note that charge-system makes that even this further relaxation remains interesting to be studied. Note finally that analogous charging-models can be assumed for the on-line model of Section 2. As it is shown in [6], results obtained are completely similar.

In what follows, we denote by t the number of steps needed that the final graph is fixed, by n the order of the final graph G , by G_i the i th cluster, by n_i the order of G_i and by H_i the part of G known at step i , i.e., the subgraph of G induced by vertices arrived during steps $1, \dots, i$ ($H_t = G$). If A is an on-line algorithm, then $A(H_i)$ will denote the solution built by A up to i th iteration.

3.1 First irrevocability relaxation

As previously in Section 2.2, we will use a threshold algorithm. Such a use is due to the “blindness” of the algorithm entailed by the fact that choices are irrevocable, at least dealing with inclusions of vertices in the solution. This means that, once on-line

algorithm A makes a choice, this choice can be fatal since the way the rest of the graph is revealed can forbid it from making any other extension of the solution it is constructing.

We have already mentioned that the charge-system we have considered, makes that the graph where A works can be assumed weighted as it has been described just previously. Consider then an off-line algorithm A solving (the off-line version of) WEIGHTED MAX INDEPENDENT SET and the following on-line algorithm, denoted by WOLTA, where threshold $r(n,t,k)$ will be precisely specified later:

- set $i = 1$;
- while $i \leq t$ and $\text{val}(A(H_i)) < r(n,t,k)$, set $i = i + 1$;
- if $i \leq t$ output $A(H_i)$, else output a vertex of G_t .

Proposition 2 *If A achieves approximation ratio $\rho(n)$ for WEIGHTED MAX INDEPENDENT SET, then WOLTA achieves (under the model assumed) competitive ratio bounded below by*

$$\sqrt{\frac{\rho(n)}{n \left(t - \frac{t-1}{k}\right)}}$$

Proof. If WOLTA outputs a solution before iteration t of graph revealing, then it guarantees competitive ratio

$$\frac{\text{val}(\text{WOLTA}(G))}{\alpha(G)} \geq \frac{r(n,t,k)}{n} \tag{5}$$

Otherwise, for any iteration, $\text{val}(\text{WOLTA}(H_i)) \leq r(n,t,k)$. Since A guarantees approximation ratio $\rho(n)$,

$$\frac{\text{val}(A(H_i))}{\alpha_w(H_i)} \geq \rho(|H_i|) \geq \rho(n) \tag{6}$$

where $\alpha_w(H_i)$ denotes the weighted stability number (i.e., the cardinality of a maximum-weight independent set) of H_i .

Let S^* be a maximum independent set of G ($|S^*| = \alpha(G)$) and consider integer sequence $a_i = |S^* \cap V(G_i)|$; obviously, $S^* \cap V(H_i)$ is an independent set of H_i , i.e., a feasible solution of WEIGHTED MAX INDEPENDENT SET on H_i . The value of this solution is at most the optimal one, i.e., $a_i + (a_{i-1}/k) + \dots + (a_1/k^{i-1}) \leq \alpha_w(H_i)$. We so get, for all i :

$$\begin{aligned} r(n,t,k) &\geq \text{val}(A(H_i)) \geq \rho(n)\alpha_w(H_i) \\ &\geq \rho(n) \left(a_i + \frac{a_{i-1}}{k} + \dots + \frac{a_1}{k^{i-1}} \right) \\ a_i + \frac{a_{i-1}}{k} + \dots + \frac{a_1}{k^{i-1}} &\leq \frac{r(n,t,k)}{\rho(n)} \end{aligned} \tag{7}$$

From inequalities of (7) we can show that:

$$\alpha(G) = a_t + a_{t-1} + \dots + a_1 \leq \left(t - \frac{t-1}{k}\right) \frac{r(n,t,k)}{\rho(n)} \quad (8)$$

(the proof of (8) is given just after the end of the current proof). Using (6), we get from (8) a competitive ratio:

$$\frac{\text{val}(\text{WOLTA}(G))}{\alpha(G)} \geq \frac{\rho(n)}{\left(t - \frac{t-1}{k}\right) r(n,t,k)} \quad (9)$$

Ratio given by 5 is increasing with $r(n,t,k)$, while the one given by 9 is decreasing with $r(n,t,k)$. Equality of both ratios holds for

$$r(n,t,k) = \sqrt{\frac{n\rho(n)}{t - \frac{t-1}{k}}}$$

In this case, the competitive ratio achieved by WOLTA is as claimed. ■

We now prove inequality in (8). Using (7), we will show that $\forall j \in \{0, 1, \dots, t-1\}$:

$$a_t + a_{t-1} + \dots + a_{j+1} + \frac{a_j}{k} + \frac{a_{j-1}}{k^2} + \dots + \frac{a_1}{k^j} \leq \left(t - j - \frac{t-j-1}{k}\right) \frac{r(n,t,k)}{\rho(n)} \quad (10)$$

Inequality in (10) is true for $j = t-1$; it is indeed inequality of (7) taking $i = t$. Suppose (10) true for $j > 0$; take also (7) for $i = j$ and multiply it by $1 - 1/k$; sum the result of the operation on (7) with (10). Then,

$$\begin{aligned} & a_t + a_{t-1} + \dots + a_{i+1} + a_j \left(\frac{1}{k} + 1 - \frac{1}{k}\right) + \\ & + a_{j-1} \left(\frac{1}{k^2} + \left(1 - \frac{1}{k}\right) \frac{1}{k}\right) + \dots + a_1 \left(\frac{1}{k^j} + \left(1 - \frac{1}{k}\right) \frac{1}{k^{j-1}}\right) \\ & \leq \frac{r(n,t,k)}{\rho(n)} \left(t - j - \frac{t-j-1}{k} + 1 - \frac{1}{k}\right) \end{aligned}$$

that is exactly (10) in range $j-1$; this inequality is true for any j . We get (8) taking $j = 0$ and the proof is complete. Note that competitive ratio obtained in Proposition 2 is slightly better than ratio $\sqrt{\rho(n)/(nt)}$ obtained (without any relaxation) in [3]. However, both ratios remain of the same order.

Remark also that algorithm consisting of waiting until the whole of graph is revealed before running A on it, trivially guarantees competitive ratio $\rho(n)/k^{t-1}$ since vertex-weights are at least equal to $1/k^{t-1}$. Since n , k and t are known in advance, one has

just, before running any algorithm, to compute values of $\rho(n)/k^{t-1}$ and of the ratio claimed by Proposition 2 and to run the algorithm associated to the best of these two values. Consequently, the following corollary holds and concludes the section.

Corollary 3 *If A is an off-line approximation algorithm for WEIGHTED MAX INDEPENDENT SET, achieving approximation ratio $\rho(n)$, then there exist an on-line algorithm for the model considered achieving competitive ratio at least*

$$\max \left\{ \frac{\rho(n)}{k^{t-1}}, \sqrt{\frac{\rho(n)}{n \left(t - \frac{t-1}{k} \right)}} \right\}$$

3.2 Further relaxation

In Section 3.1, we have relaxed irrevocability, allowing the algorithm to enter in the solution it constructs vertices arrived during previous iterations. In this section, we further can also remove from the current solution vertices entered it during former iterations. The charge-system considered here remains the same as in Section 3.1. In what follows, we still use notations introduced previously.

Consider the following algorithm, denoted by BOLA and using an off-line algorithm A solving WEIGHTED MAX INDEPENDENT SET:

- set $r = 0$;
- for $i = 1$ to t : if $r < \text{val}(A(H_i))$, then set: $S = A(H_i)$, $r = \text{val}(A(H_i))$ and $i = i + 1$;
- output S .

In fact the work of BOLA amounts in determining an independent set for any H_i and in returning the best among them, i.e.,

$$\text{val}(\text{BOLA}(G)) = \max \left\{ \frac{\text{val}(A(H_1))}{\alpha(G)}, \dots, \frac{\text{val}(A(H_t))}{\alpha(G)} \right\}$$

Let S^* be a maximum independent set of G . Set $a_i = |S^* \cap V(G_i)|$ and let b_i be the value of $S^* \cap V(H_i)$ in the weighted graph H_i ; then, $b_i = a_i + (a_{i-1}/k) + (a_{i-2}/k^2) + \dots + (a_1/k^{i-1})$.

Theorem 5 *If A achieves approximation ratio $\rho(n)$ for WEIGHTED MAX INDEPENDENT SET, then BOLA achieves competitive ratio $\rho(n)/(t - ((t-1)k))$.*

Proof. For any $i \in \{1, 2, \dots, t\}$:

$$\frac{\text{val}(A(H_i))}{\alpha_w(H_i)} \geq \rho(|H_i|) \geq \rho(n) \quad (11)$$

Obviously, $S^* \cap V(H_i)$ is a feasible solution of WEIGHTED MAX INDEPENDENT SET on H_i ; so:

$$\alpha_w(H_i) \geq \text{val}(S^* \cap V(H_i)) = b_i \quad (12)$$

Using (11) and (12) and the way final solution is built by BOLA, we get:

$$\text{val}(\text{BOLA}(G)) \geq \text{val}(\mathbf{A}(H_i)) \geq \rho(n)b_i = \rho(n) \left(a_i + \frac{a_{i-1}}{k} + \dots + \frac{a_1}{k^{i-1}} \right) \quad (13)$$

Using (13) and the same arguments as for the proof of inequality in (8), one immediately reaches:

$$\left(t - \frac{t-1}{k} \right) \text{val}(\text{BOLA}(G)) \geq \rho(n) (a_1 + a_2 + \dots + a_t) = \rho(n)\alpha(G)$$

that directly deduces the result claimed. ■

From Theorem 5, one easily sees that relaxation admitted in this section improves largely result of Proposition 2. For instance, if t is a fixed constant, BOLA reaches, despite of charges, competitive ratio of the same order as the approximation ratio of the off-line algorithm \mathbf{A} that uses as a sub-routine.

Proposition 3 *If $n \geq t(t+1)/2$, then no on-line algorithm can achieve competitive ratio (for the model dealt) better than $1/t(1 - (1/k))$.*

Proof. Let \mathbf{A} be any on-line algorithm. Consider the following way of revealing G :

- in first iteration, one reveals a clique on t vertices, in the second one a clique of size $t-1$ and so on until the $(t-1)$ th iteration where a clique of size 2 is revealed; the t th cluster will be a clique on the $n - (t(t-1)/2)$ remaining vertices;
- in iteration i , we link vertices of $\mathbf{A}(H_{i-1})$ (i.e., the ones chosen by \mathbf{A} in iteration $i-1$) to all of the vertices of the clique revealed in step i .

We show that, in any iteration, $\text{val}(\mathbf{A}(H_i)) \leq 1 + 1/k + \dots + 1/k^{i-1}$. Indeed, $\mathbf{A}(H_i)$ contains at most one vertex in any cluster since these clusters are cliques. Furthermore, upon the arrival of cluster i any vertex in $\mathbf{A}(H_{i-1})$ is linked with $V(G_i)$. So, if \mathbf{A} chooses a vertex in $V(G_i)$, then $\mathbf{A}(H_{i-1}) \cap \mathbf{A}(H_i) = \emptyset$. Since vertices of j th cluster have weight $1/k^{i-j}$, we deduce the relation claimed. So,

$$\text{val}(\mathbf{A}(G)) \leq 1 + \frac{1}{k} + \dots + \frac{1}{k^{t-1}} \leq \frac{1}{1 - \frac{1}{k}} \quad (14)$$

Now, it suffices to note that

$$\alpha(G) = t \quad (15)$$

Indeed, cluster (clique) arrived in iteration i has size $t+1-i$. and we link at most one vertex of cluster i to any vertex of clusters $j > i$. So, in any cluster, there exists at least a vertex v not linked to vertices of the subsequent clusters. The set of all these vertices v forms an independent set of cardinality t and (15) is true. Combining (14) and (15) we get: $\text{val}(\mathbf{A}(G))/\alpha(G) \leq 1/(t(1 - (1/k)))$, qed. ■

From Theorem 5 and Proposition 3, one can see that BOLA, although simple is quite competitive since, considering an optimal off-line algorithm instead of \mathcal{A} , its competitive ratio becomes $1/(t(1 - (1/k)) + 1/k)$ that is very close to upper bound given by Proposition 3.

4 Conclusion

We have presented new models for on-line MAX INDEPENDENT SET. In addition to results themselves, methods used are interesting per se since they exhibit how on-line computation can be seen as extension of polynomial approximation theory. In particular,

- algorithms devised are, for the most of them, very competitive, since their competitive ratios match upper bounds provided for the models dealt;
- competitive analyses take advantage of existing approximation results and hence, they can be seen as reductions from approximation to on-line framework.

Two major open directions that studies as the ones of the paper address are: first, the development of opposite-sense reductions, i.e. reductions from the on-line to the approximation framework and, second, development of reductions between on-line models for the same or, mainly, for distinct combinatorial optimization problems.

Références

- [1] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo. Algorithms for the on-line traveling salesman problem. *Algorithmica*, 29(4):560–581, 2001.
- [2] C. Berge. *Graphs and hypergraphs*. North Holland, Amsterdam, 1973.
- [3] M. Demange, X. Paradon, and V. Th. Paschos. On-line maximum-order induced hereditary subgraph problems. In V. Hlaváč, K. G. Jeffery, and J. Wiedermann, editors, *SOFSEM 2000—Theory and Practice of Informatics*, volume 1963 of *Lecture Notes in Computer Science*, pages 326–334. Springer-Verlag, 2000.
- [4] M. Demange and V. Th. Paschos. Improved approximations for maximum independent set via approximation chains. *Appl. Math. Lett.*, 10(3):105–110, 1997.
- [5] M. Demange and V. Th. Paschos. Improved approximations for weighted and unweighted graph problems. *Theory of Computing Systems*, 2004. To appear.
- [6] B. Escoffier. Problème *on-line* du stable de cardinalité maximale. Mémoire de DEA, 2002.
- [7] M. M. Halldórsson. Approximations via partitioning. JAIST Research Report IS-RR-95-0003F, Japan Advanced Institute of Science and Technology, Japan, 1995.

- [8] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appl.*, 4(1):1–16, 2000.
- [9] M. M. Halldórsson, K. Iwama, S. Miyazaki, and S. Taketomi. Online independent sets. *Theoret. Comput. Sci.*, 289(2):953–962, 2002.
- [10] D. S. Hochbaum, editor. *Approximation algorithms for NP-hard problems*. PWS, Boston, 1997.
- [11] V. Th. Paschos. A survey about how optimal solutions to some covering and packing problems can be approximated. *ACM Comput. Surveys*, 29(2):171–209, 1997.