ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 332 (2005) 83–108
Theoretical Computer Science
www.elsevier.com/locate/tcs

On-line vertex-covering☆

Marc Demange[a, b], Vangelis Th. Paschos[c,*]

[a] *Department of DIS, ESSEC, Avenue Bernard Hirsh, BP 105, 95021 Cergy Pontoise Cedex, France*
[b] *CERMSEM, Université Paris I, France*
[c] *LAMSADE CNRS UMR 7024, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France*

Received 8 April 2003; received in revised form 5 May 2004; accepted 19 August 2004
Communicated by G. Ausiello

Abstract

We study the minimum vertex-covering problem under two on-line models corresponding to two different ways vertices are revealed. The former one implies that the input-graph is revealed vertex-by-vertex. The second model implies that the input-graph is revealed per clusters, i.e. per induced subgraphs of the final graph. Under the cluster-model, we then relax the constraint that the choice of the part of the final solution dealing with each cluster has to be irrevocable, by allowing backtracking. We assume that one can change decisions upon a vertex membership of the final solution, this change implying, however, some cost depending on the number of the vertices changed.

Keywords: On-line algorithm; Competitive ratio; Approximation; Graph; Vertex-covering

1. Introduction

On-line computation is very natural in real world applications since it represents natural situations where the final data-set is not a priori known; in other words, data are revealed step-by-step. Frequently, when one tries to solve problems issued from such situations,

☆ An extended abstract of this paper appears in the proceedings of the 28th International Workshop on Graph Theoretical Concepts in Computer Science, WG'02, LNCS 2573.

* Corresponding author.
E-mail address: paschos@lamsade.dauphine.fr (V.Th. Paschos).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.08.015

many types of constraints (for example, deadlines on the final solution delivery, deadlines on the implementation of the computed solution) force her/him to start solving the problem before the whole set of data is completely revealed. On the other hand, these constraints may be strict enough forcing so the problem solver to irrevocably decide on the part of the final solution dealing with each part of data revealed, or may be relatively weak, allowing her/him to go back over decisions previously taken about the partial solution computed at each step.

Let Π be an **NP** optimization graph-problem. The *on-line* version of Π, denoted by ON-LINE Π, is the pair $(\Pi, \boldsymbol{R})$ where $\boldsymbol{R}$ is a set of rules dealing with

1. information on the value of some parameters of the final graph,
2. how the final graph is revealed.

An *on-line* algorithm A decides at each step which of the data (vertices or edges) revealed during this step will belong to the final solution. Its performance is measured in terms of the so-called competitive ratio c_{A} defined, for an instance G and a set $\boldsymbol{R}$, as the ratio of the worst (over all the ways G is revealed according to $\boldsymbol{R}$) value of the solution computed by it when running on G to the value of a solution computed off-line, i.e. by an algorithm running once the final graph is completely known. In this paper we deal with deterministic on-line algorithms. The notion of competitive ratio has been originally introduced in [13], in order to study a fundamental computer science problem, the paging problem. Since then, intensive research has been conducted on on-line versions of several combinatorial optimization problems. The interested reader can be referred to [6,10] for more details on theoretical and computational aspects of on-line computation. Also, an interesting financial application of on-line computing is solved in [5]. On-line graph problems studied until now are, to our knowledge, the traveling salesman [1], the graph-coloring [7,9,11] and the independent set [4]. Finally, another on-line model for independent set is studied in [8]. There, a kind of revocability on the construction of the solution is allowed by maintaining a number of alternative solutions and by choosing the best among them at the end of the game. However, no penalty is considered for such revocability.

Let us consider a company receiving manufacturing orders from its clients. These orders have to be accepted or rejected as soon as they arrive. Acceptance or rejection of the orders can be provided either immediately, i.e. as soon as any order arrives (*alternative* 1), or at the end, say of each month (*alternative* 2.1), or after a fixed number of orders (say 100 orders) has arrived (*alternative* 2.2). Incompatibilities between orders (due to the production time, the materials required, etc.) are pairwise conflicts; so a set of globally compatible orders is an independent set in the order–conflict graph (where orders are its vertices). If the objective of the company is to maximize the number of the compatible orders accepted during, say one year's period (or the global profit implied by them), then the problem to be solved is the *on-line maximum independent set problem*. However, assume a public company or even a company operating with privileged clients. In both situations, the company is constrained to accept any order emanating from its clients and has either to manufacture orders by itself, or to use sub-contractors. Then, its objective is not to maximize the profit, but to minimize the subcontracting cost during, for example, one year's period. So, with respect to the conflict graph mentioned just above, one has to minimize the complement of a maximum independent set, i.e. a vertex-cover, and the problem to solve is the *on-line minimum vertex-covering*. This is the problem we deal with in this paper.

The minimum vertex-covering problem, denoted by VC in the sequel, is defined as follows: *given a graph* $G(V, E)$, *compute the minimum-cardinality set* $V' \subseteq V$ *such that,* $\forall v_i v_j \in E$, *at least one of the* v_i, v_j *belongs to* V'. We consider that G (we set $n = |V|$ and suppose n known at the beginning of the game) is revealed per non-empty clusters, i.e. per induced vertex–disjoint subgraphs $G_1(V_1, E_1), G_2(V_2, E_2), \ldots$ of G (we denote by n_i the size of V_i, $i = 1, 2, \ldots$). Every time a new cluster G_i is revealed, the edges linking the vertices of G_i with the vertices of G_j, $j < i$ are also revealed. We denote by t the number of clusters needed so that the whole graph is completely revealed.

We first focus ourselves on the case where the graph is revealed by means of its vertices and consider $t = n$, i.e., that G is revealed vertex-by-vertex. This is what we have called *alternative* 1 in our company model described previously. We establish a general result about the performance of every minimal VC-algorithm (i.e., an algorithm computing a minimal vertex-cover) in comparison with the maximum matching algorithm for VC, informally, the ratio of any minimal vertex-cover against the vertex-cover computed by the maximum matching algorithm is bounded above by $\Delta/2$, where Δ is the maximum degree of the final graph. Using this result, we establish the competitive ratio of a very simple but very natural on-line algorithm entering a newly presented vertex v in the covering C under construction, if there exists an edge incident to v (hence revealed together with v) the already revealed endpoint u of which does not belong to C (following our assumption about the way G is revealed, u has arrived before v).

Next, we generalize our study assuming $t < n$ and study the competitive ratio of (more complicated) on-line algorithms for ON-LINE VERTEX COVER against an optimal off-line algorithm. Here we distinguish two cases: $2 < t < n$ and $t = 2$. With respect to our company model, the former represents *alternative* 2.1; *alternative* 2.2, not studied here, could represent a situation where all the clusters are of the same order. Then, we analyze the case $t = 2$. This case has, as we shall see, its own mathematical interest. Furthermore, even in the framework of our company model it is very natural. Revisit this model and suppose that in order that the products start to be manufactured, say at the instant $t + 15$, orders have to be arrived at instant t. But for some reasons (e.g., organizational or promotional ones), clients have been granted some extra delay, for example $t + 10$, in order to send orders to be manufactured at $t + 15$. Here, company has to answer in two times: first, for the orders arrived up to instant t and second, for the orders arrived from $t + 1$ to $t + 10$. The conflict graphs of these two sets of orders are the two clusters.

We continue the paper by assuming non-irrevocability in the construction of the on-line solution, i.e., by allowing backtracking. This means that the algorithm can interchange a number of vertices in the solution computed by a number of vertices not included in it. But we consider that changes performed imply a cost. This, in our company example, becomes in deciding, at the last moment, to give some additional manufacturing work in its subcontractors. But since it does not meet the deadline for these orders, it has to pay some extra cost. We study the competitiveness (against an optimal off-line algorithm) of two algorithms under a general cost-model. This assumes that the cost paid for the change of the status of x vertices is $f(x)$ for a positive non-decreasing function f.

Finally, we study a slightly different on-line model, where we assume that the input-graph is revealed edge-by-edge. Together with the arrival of a new edge, are revealed the links of its endpoints with the ones of the edges already revealed. Here also we

Table 1
Summary of the results of Sections 3–5

t	Upper bounds	Lower bounds
$t = n$	Δ	Δ[a]
$2 < t < n$	Δ[b]	$\Delta - 2$[c]
$t = 2$	$(\Delta + 5)/2$[d]	$(\Delta + 1)/2$[e]
G is revealed edge-by-edge	2[f]	

[a] The bound holds even if a graph isomorphic of G is known in advance; this bound becomes $\Delta - 2$ if G is a tree and n is known in advance (Section 3).

[b] This ratio becomes $(\Delta + 1)/2$ when assuming that clusters arrive without isolated vertices (Section 4).

[c] Assuming that $t = \Theta(\sqrt{n} \log n)$; this bound holds even if the input graph is a tree, any cluster is non-empty and n is known in advance; for any other value of t, question is open (Section 4).

[d] Assuming that the final graph has no isolated vertices; assuming, furthermore, that clusters arrive without such vertices, the ratio becomes asymptotically $\Delta/2.62$ (Section 4).

[e] The bound holds even if the final graph is bipartite, has no isolated vertices, both clusters have the same size and a graph isomorphic to the input graph is known in advance (Section 4).

[f] This bound is tight (Section 5).

devise an on-line algorithm and study its competitive ratio against an optimal off-line one.

The overall purpose of the paper is of course to study several on-line models, but also to exhibit links between polynomial approximation and on-line computation for ON-LINE VERTEX COVER. It is well known that VC belongs to **APX** (the class of problems approximable within constant approximation ratio) since the maximum matching algorithm achieves approximation ratio 2 for it. Our way to process here is to study competitive ratios of natural and simple on-line algorithms using maximum matching computations as basic operations. As we will see, use of such computations in our models does not lead to “good” competitive ratios since they are all of order of the maximum degree of the final graph and, moreover, for any upper bound proved, we simultaneously provide lower bounds of the same order. This is fairly strange since it exhibits a dissymmetry with respect to the problems studied in [4]. There, under very similar models, it is shown that when a non-trivial off-line approximation algorithm is used as basic part of an on-line independent set algorithm, the competitive ratios achieved are only by a logarithmic factor inferior to the off-line approximation ratio.

Finally, let us note that some of the hypotheses adopted in order to derive some of the results of the paper do not seem very natural. We speak about Corollary 3 of Section 4.1 and Theorem 4 of Section 4.2.2, where the basic assumption made is that clusters arrive without isolated vertices. Despite this drawback concerning these two results, we have decided to analyze the corresponding cases in order to show how additional hypotheses on the structure of the instances influence on competitive results.

Table 1 summarizes the main results of Sections 3–5. Due to several hypotheses on the cost-model considered in Section 6 dealing with backtracking, it is quite complicated to summarize the results of this section; therefore, they are omitted from Table 1.

2. Basic definitions and notations

The following definitions will be frequently used throughout the paper. For reasons of readability they are grouped here, before entering the purely technical part.

Matching: A matching is a set of mutually disjoint edges of G.

Exposed vertices: A vertex is called exposed with respect to a matching M, if it is not endpoint of any edge of M, in other words, if it is not saturated by M.

Augmenting path: A path $v_{i_1}, v_{i_2}, \dots, v_{i_k}$ is augmenting with respect to a maximal matching M if k is even, v_{i_1}, v_{i_k} are exposed with respect to M and if $v_{i_l}v_{i_{l+1}} \in M$, $l = 2q$, $q = 1, \dots, (k-2)/2$; in other words, the set $M \setminus \{v_{i_l}v_{i_{l+1}} : l = 2q, q = 1, \dots, (k-2)/2\} \cup \{v_{i_{l+1}}v_{i_{l+2}} : l = 2q, q = 0, \dots, (k-2)/2\}$ is also a matching with cardinality equal to $|M| + 1$.

Independent set: An independent set is a subset of $V' \subseteq V$ such that, for any $(v_i, v_j) \in V' \times V'$, $v_i v_j \notin E$.

Minimal (resp., maximal) set: A set will be called minimal (resp., maximal) with respect to a property π, if it satisfies π, while deletion (resp., insertion) of an element from (resp., in) S results in a set not satisfying π.

Fact 1 (*Berge* [3]). *Any (maximal) independent set is the complement, with respect to V, of a (minimal) vertex-cover*.

In what follows, for $v \in V$, we denote by $\Gamma(v)$ the set of neighbors of v, i.e., $\Gamma(v) = \{u : uv \in E\}$; we denote by Δ the maximum degree of G, i.e., $\Delta = \max\{|\Gamma(v)| : v \in V\}$. By $\tau(G)$ we denote the cardinality of a minimum vertex cover of G and by M (resp. M_i) a maximum matching of G (resp., G_i) and by P (resp. P_i) the exposed vertices of G (resp., G_i) with respect to M (resp., M_i), i.e., the vertices of G (resp., G_i) not saturated by M (resp., M_i). We will also denote by $X(M)$ (resp., $X(M_i)$) the set of the endpoints of M (resp., M_i).

Fact 2 (*Berge* [3]). *Consider a graph G, fix a maximal matching M and set $|M| = m$. Then*
1. *$V \setminus X(M)$ is independent for G;*
2. *$X(M)$ is a vertex-cover of G with $|X(M)| = 2m$.*

3. On-line vertex-covering with $t = n$

We first consider that G is revealed into $t = n$ clusters, i.e., vertex-by-vertex. Before specifying an on-line algorithm for this case, we establish a general result for any algorithm (on-line or off-line) computing a minimal vertex-cover, i.e., a vertex-cover that cannot be reduced by elimination of some of its vertices.

3.1. *On the approximation ratio of any minimal vertex-covering algorithm against maximum matching*

We denote by `MAX_MATCHING` an algorithm computing a maximum matching M of G (the problem of finding a maximum matching of a graph is polynomial [12]). By Item 2 of Fact 2, $X(M)$ is a vertex-cover of size $2|M|$ for G. Set $m = |M|$ and $p = |P|$. The items of the following easy lemma will be frequently used in the sequel.

Lemma 1. *Consider a graph $G(V, E)$ with n vertices and denote by M a maximum matching of G, by $X(M)$ the set of the endpoints of the elements of M, by m the cardinality of M and by p the cardinality of the set $P = V \setminus X(M)$. Then,*
1. *for any graph without isolated vertices, $p \leqslant m(\Delta - 1)$; if, in addition, the graph contains $\imath$ isolated vertices, then $p - \imath \leqslant m(\Delta - 1)$;*
2. *in any graph with no isolated vertices, $m \geqslant n/(\Delta + 1)$; if the graph has $\imath$ isolated vertices, then $m \geqslant (n - \imath)/(\Delta + 1)$;*
3. *for any graph G, the ratio of the size of any minimal vertex-cover to the size of the vertex-cover induced by `MAX_MATCHING(G)` is bounded above by $\Delta/2$.*

Proof of Item 1. Fix an edge $v_i v_j \in M$ such that at least one of v_i, v_j has neighbors in $V \setminus X(M)$. If both of them have such neighbors, then observe that there cannot exist two distinct exposed vertices *x* and *y* such that v_i is linked to one of them, say *x*, and v_j to *y*. Otherwise, there would be an augmenting path with respect to *M* contradicting the maximality of *M*. Consequently, $|(\Gamma(v_i) \cup \Gamma(v_j)) \cap (V \setminus X(M))| \leqslant \Delta - 1$, since $v_i v_j$ is an edge of *E* and contributes by one unit to the degrees of v_i and v_j. Iterating this argument for any edge of *M* we get the result claimed.

If *G* contains a set *I* of $\imath$ isolated vertices, then the argument developed above remains valid on the graph $G'(V \setminus I, E)$, q.e.d.

Proof of Item 2. Since $n = 2m + p$ and, by Item 1, $p \leqslant m(\Delta - 1)$ (resp., $p - \imath \leqslant m(\Delta - 1)$), one easily gets $n \leqslant m(\Delta + 1)$ (resp., $n - \imath \leqslant m(\Delta + 1)$) and reaches the result.

Proof of Item 3. The claim clearly holds for $\Delta = 1$. Suppose $\Delta \geqslant 2$. Consider a minimal vertex-cover *C* and denote by M' the subset of *M* for any edge of which both endpoints belong to *C*. Remark first that $V \setminus C$ is a maximal independent set. Remark also that, by Item 1 of Fact 2, the set $K = C \setminus X(M)$ is independent. We claim that vertices in *K* receive edges from saturated vertices in $V \setminus C$. In fact, existence of an edge between a vertex in *K* and a vertex in $V \setminus X(M)$ contradicts the maximality of *M*. Next, existence of an edge between a vertex in *K* and a vertex in $X(M) \cap C$ contradicts the minimality of *C*. Note finally that the number of saturated vertices in $V \setminus C$ is equal to $m - |M'|$. Henceforth, $|K| \leqslant (m - |M'|)(\Delta - 1)$. We so have, $|C| = 2|M'| + m - |M'| + |K| \leqslant \Delta m = (\Delta/2)2m$ since $\Delta \geqslant 2$ and the claim follows. □

It is well-known [3] that, for any graph *G* and for any maximal matching *M* (of cardinality *m*) of *G*

$$\tau(G) \geqslant m. \tag{1}$$

Consequently, by Item 3 of Lemma 1, the following corollary holds immediately.

Corollary 1. *For any graph, the ratio of the size of any minimal vertex-cover to the size of the optimal one is bounded above by Δ.*

3.2. An on-line algorithm for the case $t = n$

We analyze here a very simple but very natural on-line algorithm, called `OLVC` in what follows, entering a newly presented vertex v in the covering C under construction, whenever there exists an edge incident to v (hence revealed together with v) the already revealed endpoint u of which does not belong to C (following our assumption about the way G is revealed, u has arrived before v).

Proposition 1. *The competitive ratio of* `OLVC` *against an optimal off-line algorithm for VC is bounded above by Δ. This bound is tight.*

Proof. Following `OLVC`, $\forall v \in C$, $\exists uv \in E$ such that $u \notin C$. Hence, the vertex-cover C computed is minimal. Then, application of Corollary 1 concludes the ratio claimed.

Fix now a $\Delta \in \mathbb{N}$, consider a star $S_{\Delta+1}$ on $\Delta + 1$ vertices. Obviously, $\tau(S_{\Delta+1}) = 1$. Suppose that its center is the first vertex revealed; the rest of vertices can be revealed in any order. Then, `OLVC` will not include the star-center in C, while it will include all the remaining vertices of $S_{\Delta+1}$. Therefore, the competitive ratio achieved in this case is equal to Δ. $\square$

Let us note that Proposition 1 can be proved by the following straightforward arguments. Since C is minimal, for any connected component of G, there exists at least one edge covered by only one element of C; hence, $|C| \leqslant |E|$. On the other hand, since any vertex can cover at most Δ edges, the size of any cover, even of an optimal one, is at least $\lceil |E|/\Delta \rceil$. The competitive ratio follows.

3.3. Lower bounds on the competitiveness of any algorithm for the case $t = n$

Suppose that vertices are numbered in the order they arrive; in step i, vertex v_i is revealed. Also consider that, in step i, $\{v_1, \ldots, v_i\} = C_i \cup S_i$, where C_i draws the vertex-set included in the vertex cover under construction and $S_i = \{v_1, \ldots, v_i\} \setminus C_i$. The final graph is denoted, as usual, by $G(V, E)$ and its maximum degree by Δ. The purpose of this section is to provide limits for the competitiveness (against an optimal off-line algorithm) of any on-line algorithm solving ON-LINE VERTEX COVER with $t = n$ (over all the ways the input-graph is revealed). Let us consider the solution of ON-LINE VERTEX COVER as a two-players game, where the first one (Player 1) reveals the instance and the second one (Player 2) constructs the solution. Then, we prove the following theorem.

Theorem 1. 1. *No algorithm can achieve competitive ratio strictly better than Δ, even if a graph G' isomorphic to G is known in advance.*
2. *No algorithm can achieve competitive ratio strictly better than $\Delta - 2$, even if G is a tree and n is known in advance.*

Proof of Item 1. The graph G', isomorphic of G, revealed in advance consists of a disjoint collection of p stars, each of order $\Delta + 1$ and of $\Delta - 1$ isolated vertices, where Δ

and p are fixed integers. Obviously, the maximum degree of G is Δ and its order $n = p(\Delta + 1) + \Delta - 1$. Assume that Player 1 reveals the graph with respect to the following rules:

i if C_i contains Δ isolated vertices (for the graph already revealed), then v_{i+1} is linked to all these vertices;

ii if $v_i \in S_i$ (in other words, v_i has not been taken in the solution) and v_i is not linked to any vertex v_j, $j < i$, and if $i \leqslant n - \Delta$, then vertices $v_i, v_{i+1}, \ldots, v_{i+\Delta}$ form a star rooted in v_i;

iii if p stars have been revealed, the rest of the vertices revealed are isolated;

iv if Rules i and ii cannot be applied and $i \leqslant n - 1$, then vertex v_{i+1} is isolated with respect to the graph already revealed.

Application of Rules i–iv above implies that Player 2 cannot do better than covering edges of any star by its leaves, while optimal off-line solution consists of the star-centers. Therefore, a ratio of Δ is achieved at best and this completes the proof of Item 1 of the theorem.

Proof of Item 2. Let Δ be an integer greater than, or equal to, 3 and set $n = \Delta(\Delta + 1) + 1$. Consider that Player 1 reveals the graph following the rules below:

(i) if C_i contains Δ isolated vertices (with respect to the graph already revealed) and $i \leqslant n - 2$, then v_{i+1} is linked to all these isolated vertices;

(ii) if $v_i \in S_i$ (in other words, v_i has not been taken in the solution) and v_i is not linked to any vertex v_j, $j < i$, and if $i \leqslant n - \Delta - 1$, then vertices $v_i, v_{i+1}, \ldots, v_{i+\Delta}$ form a star rooted in v_i;

(iii) consider $v_i \in S_i$, v_i isolated with respect to the graph already revealed, and $n - 2 \geqslant i \geqslant n - \Delta$; set $A = \{v_j : j < i, v_j \in C_i, \forall k \leqslant i, v_j v_k \notin E\}$ (i.e., A is the set of the isolated vertices, at instant i, taken in C_i) and $B = \{v_{i+2}, \ldots, v_{n-1}\}$; then:
  (a) v_{i+1} is linked to v_i and to any element of set A;
  (b) the elements of B form an independent set and are linked to v_i

(iv) if Rules (i) and (ii) do not apply and if $i \leqslant n - \Delta$, then vertex v_{i+1} is isolated with the graph already revealed;

(v) v_n is linked to Δ vertices of degree 1 picked in the several connected components of the graph revealed until step $n - 1$.

If Rule (iii) is not applied, then in step $n - 1$, the graph contains Δ stars, whose vertices of degree 1 make part of the solution constructed by Player 2. In this case, $\tau(G) = \Delta + 1$ (the roots of the stars plus vertex v_n), while the solution constructed is of size Δ^2. The competitive ratio is in this case at least $\Delta - 1$.

Suppose now that Rule (iii) is applied (recall that this is the case for vertex 27 in Fig. 1). Then in step i, the graph consists of k stars (their leaves making part of the solution constructed by Player 2) plus the vertices of $A \cup \{i\}$. In this case, the total number of vertices verifies $n = \Delta(\Delta + 1) + 1 = k(\Delta + 1) + |A| + |B| + 3$, with $|A| < \Delta$ (if not, Rule (i) would be applied one more time) and $|B| \leqslant \Delta - 2$ (because $i \geqslant n - \Delta$). We deduce $k = \Delta - 1$. In this case, $\tau(G) = \Delta - 1 + 3 = \Delta + 2$, while the solution finally constructed by Player 2 has at least $(\Delta - 1)\Delta + \Delta = \Delta^2$ vertices (the leaves of the $\Delta - 1$ stars plus set A plus set B plus v_{i+1}). The competitive ratio implied is then at least $\Delta - 2$.

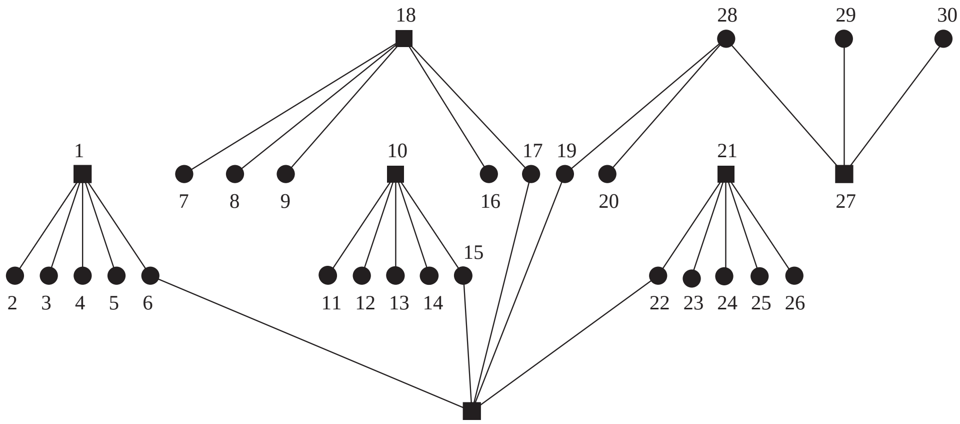

Fig. 1. A graph fitting Rules (i)–(v) with $n = 31$ and $\Delta = 5$.

In order to conclude, let us note that the graph at step $n-1$ consists of Δ acyclic connected components, each of them containing at least one vertex of degree 1. This makes that Rule (v) is feasible and guarantees that the final graph is a tree. So, the proofs of Item 2 and of the theorem are complete. □

Remark 1. With respect to Item 1 of Theorem 1, if we suppose that the final graph is a collection of p disjoint stars each of degree Δ (without isolated vertices), application of Rules i and ii yields a lower bound of $\Delta - ((\Delta - 2)/k)$. This bound is asymptotically equal to Δ.

In Fig. 1 a graph G fitting Rules (i)–(v) is shown for $n = 31$ and $\Delta = 5$. Vertices are numbered in the order they have been revealed. Here, Rule (iii) is applied for vertex 27. Then, $A = \{19, 20\}$ and $B = \{29, 30\}$. The circle vertices represent the ON-LINE VERTEX COVER-solution, while the square ones represent the independent set associated with it. A graph fitting Item 1 of the theorem could be as the one of Fig. 1 induced by the set of vertices $\{1, \ldots, 27\}$ plus one isolated square vertex. In this example, $\tau(G) = 7$. The optimal solution is the set $\{1, 10, 18, 21, 27, 28, 31\}$, while the solution constructed by Player 2 (the set of the circle vertices of Fig. 1) is of cardinality 25.

4. On-line vertex covering with $n > t \geqslant 2$

We assume in this section that G is revealed by non-empty clusters G_i, $i = 1, \ldots, t$, with $2 \leqslant t < n$. We first study the case $t > 2$. The case $t = 2$, being interesting by itself, is examined separately in Section 4.2. We suppose that t is known at the beginning of the game.

4.1. On-line vertex covering with $n > t > 2$

For the case we deal with in this section, we propose the following algorithm, denoted by `t_OLVC`:
- arrival of G_1: set $C = X(\texttt{MAX\_MATCHING(G}_1))$;
- arrival of G_i, $i = 2, \ldots, t$:
  (a) set $C = C \cup X(\texttt{MAX\_MATCHING}(\texttt{G}_\texttt{i}))$;
  (b) for any $u \in V_i \setminus X(\texttt{MAX\_MATCHING}(\texttt{G}_\texttt{i}))$, if $\exists v \in (\cup_{1 \leqslant j \leqslant i-1} V_j) \setminus C$ such that $uv \in E$, then set $C = C \cup \{u\}$;
- output C.

Obviously, the set C finally computed by `t_OLVC` is a vertex-cover, although not necessarily minimal. So, Proposition 1 does not represent the worst case for its competitive ratio. Note that for the case where clusters are assumed without any restriction, setting $|C| \leqslant n - |I|$ (where I denotes the set of isolated vertices, if any) and using Item 2 of Lemma 1, competitive ratio $\Delta + 1$ is immediately deduced.

Theorem 2. *Let λ_i be the number of the isolated vertices of G_i introduced in C and set $\lambda = \sum_{i=1}^{t} \lambda_i$. Denote by A_i, $i = 2, \ldots, t$, the vertex-sets introduced in C during the execution of Step* (b) *of* `t_OLVC`*, set $A = \cup_{i=2}^{t} A_i$, $\rho = \lambda/|A|$. Then, the competitive ratio of* `t_OLVC` *against an optimal VC algorithm is bounded above by* $2 + (\Delta - 2)/(2 - \rho)$.

Proof. Denote by M_i, $i = 1, \ldots, t$, a maximum matching of G_i and set $m_i = |M_i|$, $i = 2, \ldots, t$. Observe that vertex-set A is exposed with respect to the (non-maximum) matching $\cup_{i=1}^{t} M_i$; moreover, it does not contain any isolated vertex. Observe also that any isolated vertex is exposed with respect to any matching of G; hence

$$\lambda \leqslant |A| \Longleftrightarrow \rho \leqslant 1. \tag{2}$$

The cardinality of the on-line solution C computed by `t_OLVC` can be written as follows:

$$|C| = 2 \sum_{i=1}^{t} m_i + |A|. \tag{3}$$

Let E' be the set of edges that have entailed introduction of the vertices of A in C and denote by $B(X(E'), E')$ the partial subgraph of G defined on vertex-set $X(E')$ and on edge-set E'. Also, denote by M_B a maximum matching of B and by m_B the cardinality of M_B. Remark also that $M_B \cup_{i=1}^{t} M_i$ is a matching of G, not necessarily maximum but maximal, and that the graph B is bipartite with color-classes A and $X(E') \setminus A$. Expression (1) can, for the purposes of our proof, be rewritten as

$$\tau(G) \geqslant \left(\sum_{i=1}^{t} m_i \right) + m_B \tag{4}$$

and, using (3) and (4), the competitive ratio of `t_OLVC` becomes

$$c_{\texttt{t\_OLVC}} = \frac{|C|}{\tau(G)} \leqslant \frac{|C|}{\left(\sum_{i=1}^{t} m_i\right) + m_B} = \frac{2\sum_{i=1}^{t} m_i + |A|}{\left(\sum_{i=1}^{t} m_i\right) + m_B}$$
$$= 2 + \frac{|A| - 2m_B}{\left(\sum_{i=1}^{t} m_i\right) + m_B}. \qquad (5)$$

Set $A' = X(E') \setminus A$. Then, any vertex in $X(M_B) \cap A'$ is linked to at most Δ vertices of A. On the other hand, no vertex in $A \setminus X(M_B)$ is linked to a vertex of $A' \setminus X(M_B)$; if not, M_B would not be maximum. Consequently, since any vertex of A has at least one neighbor in A'

$$|A| \leqslant \Delta m_B. \qquad (6)$$

On the other hand, let $i \in \{1, \ldots, t\}$ and let I_i be the set of the isolated vertices of cluster G_i. For $i = 1, \ldots, t$, denote by P_i the exposed vertices of M_i with respect to V_i. We then have $|I_i| = \lambda_i + (|I_i| - |A_i \cap I_i|)$, or $|A_i| - \lambda_i = |A_i| + (|I_i| - |A_i \cap I_i|) - |I_i| \leqslant p_i - |I_i| \leqslant m_i(\Delta - 1)$, where the first inequality holds because $A_i \subseteq P_i$ and, as we have already mentioned in the beginning of the proof, the set $I_i \setminus (A_i \cap I_i)$, being isolated in G_i, is exposed with respect to any matching of G_i; the second inequality holds thanks to Item 1 of Lemma 1. Summing inequalities $|A_i| - \lambda_i \leqslant m_i(\Delta - 1)$ for $i = 1, \ldots, t$, we obtain

$$\sum_{i=1}^{t} m_i \geqslant \frac{|A| - \lambda}{\Delta - 1}. \qquad (7)$$

Using (2), (6) and (7), expression (5) becomes

$$c_{\texttt{t\_OLVC}} \leqslant 2 + \frac{|A| - 2m_B}{\left(\sum_{i=1}^{t} m_i\right) + m_B} \leqslant 2 + \frac{|A| - \frac{2|A|}{\Delta}}{\frac{|A|-\lambda}{\Delta-1} + \frac{|A|}{\Delta}}$$
$$= 2 + \frac{\frac{\Delta-2}{\Delta}}{\frac{\Delta(1-\rho)+\Delta-1}{\Delta(\Delta-1)}} = 2 + \frac{(\Delta-2)(\Delta-1)}{\Delta(2-\rho)-1} \leqslant 2 + \frac{\Delta-2}{2-\rho} \qquad (8)$$

and the proof of the theorem is complete. □

Recall that $\rho \leqslant 1$ (see (2)). Furthermore, (8) is increasing with ρ; hence, setting $\rho = 1$ the following result is immediately obtained.

Corollary 2. *The competitive ratio of `t_OLVC` against an optimal VC algorithm is bounded above by Δ.*

Note that the solution C computed by `t_OLVC` is not necessarily minimal. Consequently, the result of Corollary 2 cannot be derived by direct application of Corollary 1. The bound of Theorem 2 can be slightly improved in the case where clusters arrive without isolated vertices by the following way. Since, for $i = 1, \ldots, t$, the vertices of A_i are exposed with respect to M_i, using Item 1 of Lemma 1 ($A_i \cap I = \emptyset$), we get

$$|A_i| \leqslant m_i(\Delta - 1). \qquad (9)$$

On the other hand, since no cluster contains isolated vertices, the bipartite graph $B(X(E'), E')$ considered in the proof of Theorem 2 has maximum degree bounded above by $\Delta - 1$ (at least one edge per vertex in $X(E')$ links it to vertices of $X(M_i)$, $i = 1, \ldots, t$). Consequently, taking also into account that there exist no edges among sets $A \setminus X(M_B)$ and $A' \setminus X(M_B)$, we get

$$|A| \leqslant (\Delta - 1)m_B. \tag{10}$$

Combining (9) and (10), expression (5) becomes

$$c_{\mathtt{t\_OLVC}} \leqslant 2 + \frac{|A| - 2m_B}{\left(\sum_{i=1}^{t} m_i\right) + m_B} \leqslant 2 + \frac{|A| - 2\frac{|A|}{\Delta-1}}{\frac{2|A|}{\Delta-1}} = 2 + \frac{\Delta - 3}{2} = \frac{\Delta + 1}{2} \tag{11}$$

and (11) leads immediately to the following final corollary.

Corollary 3. *Whenever clusters arrive without isolated vertices, the competitive ratio of* `t_OLVC` *against an optimal VC algorithm is bounded above by* $(\Delta + 1)/2$.

4.2. On-line vertex-covering with $t = 2$

Suppose now that the input graph is revealed in just two clusters $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$. Assume also that n, the order of the final graph, is known at the beginning of the game. We recall that, following our assumptions, one has to decide which vertices of the first cluster will belong to the final solution before the arrival of the second cluster.

4.2.1. G has no isolated vertices

In this section we suppose that no additional hypotheses are admitted on the forms of the clusters and analyze the competitive ratio of the following algorithm, denoted by `2_OLVC`:

- arrival of G_1:
  - (i) if $|V_1| \leqslant n/2$, then set $C = V_1$ and go to Step (1);
  - (ii) if $|V_1| > n/2$, then set $C = X(\mathtt{MAX\_MATCHING}(\mathtt{G}_1))$ and go to Step (2);
- arrival of G_2:
  - (1) output $C = C \cup X(\mathtt{MAX\_MATCHING}(\mathtt{G}_2))$;
  - (2) output $C = C \cup V_2$.

Theorem 3. *If G has no isolated vertices, then the competitive ratio of* `2_OLVC` *against an optimal VC algorithm verifies* $c_{2\_OLVC} \leqslant (\Delta + 5)/2$. *This ratio is tight.*

Proof. Denote, for $i = 1, 2$, by M_i the matchings computed by `MAX_MATCHING` on G_i at steps (ii) and (1) and by m_i their sizes.

Suppose Step (i) of `2_OLVC` is executed. Then, the solution returned is $C = V_1 \cup X(M_2)$ with $|C| \leqslant n/2 + 2m_2$. Combining expression for C with (1) and taking into account Item 2 of Lemma 1 and the fact that $m_2 \leqslant m$, the following holds:

$$c_{\mathtt{2\_OLVC}} = \frac{|C|}{\tau(G)} \leqslant \frac{\frac{n}{2} + 2m_2}{m} \leqslant \frac{n}{2m} + 2 \leqslant \frac{\Delta + 1}{2} + 2 = \frac{\Delta + 5}{2}. \tag{12}$$

Suppose now that Step (ii) of `2_OLVC` is executed instead. Then, $|V_2| \leqslant n/2$ and the solution returned is the one of Step (2), i.e. $C = X(M_1) \cup V_2$. In this case also the arguments previously developed hold. Hence, (12) always gives the competitive ratio achieved.

Let us now show that the analysis above is asymptotically tight. Consider a graph $G(V, E)$ collection of R stars, each of maximum degree Δ. Consider the subgraph G_1 of G consisting of a set of $n/2$ exposed vertices with respect to a maximum matching M of G. Remark that M contains one edge per star and that $V(G_1)$ is a set of isolated vertices of size not larger than $n/2$. Set $G_2 = G[V \setminus V(G_1)]$ and assume that G is revealed per clusters G_1 and G_2. Then

$$|C| = \frac{n}{2} + 2R = \frac{n}{2} + \frac{2n}{\Delta + 1}, \tag{13}$$

$$\tau(G) = \frac{n}{\Delta + 1} = R, \tag{14}$$

$$\frac{|C|}{\tau(G)} = \frac{\Delta + 5}{2}. \tag{15}$$

This completes the proof of the theorem. □

4.2.2. Clusters have no isolated vertices

Suppose now that clusters G_i arrive with no isolated vertices. Let n_i be the order of G_i. Denote by M_i, $i = 1, 2$, a maximum matching of G_i, by P_i the set of the exposed vertices with respect to M_i, by p_i its cardinality and consider the following on-line algorithm, denoted by `C2_OLVC` and called with parameters n and a fixed constant $\varepsilon > 1$:

- arrival of G_1: if $n_1 \leqslant n/\varepsilon$, then set $C = V_1$; else set $C = X(\mathtt{MAX\_MATCHING(G_1)})$;
- arrival of G_2:
  (a) set $C = C \cup X(\mathtt{MAX\_MATCHING(G_2)})$;
  (b) set $A_2 = \{v \in V_2 \setminus X(\mathtt{MAX\_MATCHING(G_2)}) : \exists u \in V_1 \setminus C, uv \in E\}$;
  (c) output $C = C \cup A_2$.

Theorem 4. *Under the hypothesis that clusters arrive with no isolated vertices, there exists ε_0, the largest among the roots of the polynomial $\varepsilon^2 - 3\varepsilon + 1$, such that the competitive ratio of `C2_OLVC`(n, ε_0) against an optimal VC algorithm is bounded above by $2 + ((\Delta + 1)/\varepsilon_0)$.*

Proof. Set, for $i = 1, 2$, $m_i = |M_i|$ and note the following fact that can be immediately deduced from `C2_OLVC`.

Fact 3. *Whenever then-consequence of the first item is executed, then Step* (b) *computes $A_2 = \emptyset$; therefore the final covering C computed in Step* (c) *satisfies $C = V_1 \cup X(M_2)$.*

Suppose that execution of the first item is as stated in Fact 3. Then, by (1) and Item 2 of Lemma 1, we get

$$\frac{|C|}{\tau(G)} \leqslant \frac{n_1 + 2m_2}{m} \leqslant \frac{\frac{n}{\varepsilon} + 2m_2}{m} \leqslant \frac{\frac{n}{\varepsilon}}{m} + 2 \leqslant \frac{\frac{n}{\varepsilon}}{\frac{n}{\Delta+1}} + 2 = \frac{\Delta + 1}{\varepsilon} + 2. \tag{16}$$

Suppose now that the else instruction of the first item is executed instead. Then, the set C finally computed in Step (c) by C2_OLVC verifies

$$|C| = |X(M_1)| + |X(M_2)| + |A_2| = 2(m_1 + m_2) + |A_2|. \tag{17}$$

Denote by $Q_1 \subseteq V_1 \setminus X(M_1)$ the set of vertices of V_1 that has entailed the introduction of set A_2 in C, and by $B(Q_1, A_2, E_B)$, the subgraph of G induced by $Q_1 \cup A_2$. Since they are both independent (subsets of P_1 and P_2, respectively), B is bipartite. Denote also by M_B a maximum matching of B and set $m_B = |M_B|$. Since $M_1 \cup M_2 \cup M_B$ is a maximal matching for G

$$\tau(G) \geqslant m_1 + m_2 + m_B. \tag{18}$$

Consider set $X(M_B) \cap Q_1$; obviously, $|X(M_B) \cap Q_1| = m_B$. Since G_1 is supposed without isolated vertices, any vertex of $X(M_B) \cap Q_1$ has at most $\Delta - 1$ neighbors in A_2. On the other hand, M_B being maximum for B, any vertex of A_2 receives edges from at least one vertex of $X(M_B) \cap Q_1$. So

$$|A_2| \leqslant m_B(\Delta - 1). \tag{19}$$

Also, since G_1 and G_2 are both assumed without isolated vertices, application of Item 2 of Lemma 1 gives

$$\begin{aligned} m_1 &\geqslant \tfrac{n_1}{\Delta+1}, \\ m_2 &\geqslant \tfrac{n_2}{\Delta+1}. \end{aligned} \tag{20}$$

Combining (17)–(20), performing some little and easy algebra and taking into account $n_1 + n_2 = n$, one gets

$$\frac{|C|}{\tau(G)} \leqslant \frac{2(m_1 + m_2) + |A_2|}{m_1 + m_2 + m_B} = 2 + \frac{|A_2| - 2m_B}{m_1 + m_2 + m_B} \leqslant 2 + \frac{\frac{\Delta-3}{\Delta-1}|A_2|}{\frac{n}{\Delta+1} + \frac{|A_2|}{\Delta-1}}. \tag{21}$$

Recall that we are currently considering case $n_1 > n/\varepsilon$, i.e.

$$n_2 < n - \frac{n}{\varepsilon} = n\frac{\varepsilon - 1}{\varepsilon}. \tag{22}$$

Using (20) for m_2, denoting by p_2 the number of the exposed vertices of V_2 with respect to M_2, and using (22), we obtain

$$|A_2| \leqslant p_2 = n_2 - 2m_2 \leqslant n_2 - \frac{2n_2}{\Delta + 1} = \frac{\Delta - 1}{\Delta + 1} n_2 \leqslant \frac{(\Delta - 1)(\varepsilon - 1)}{(\Delta + 1)\varepsilon} n. \tag{23}$$

Remark also that (21) is increasing with $|A_2|$. So, combining (21) and (23), we get

$$\frac{|C|}{\tau(G)} \leqslant 2 + \frac{\frac{(\Delta-3)(\varepsilon-1)}{(\Delta+1)\varepsilon} n}{\frac{n}{\Delta+1} + \frac{(\varepsilon-1)n}{\varepsilon(\Delta+1)}} = 2 + \frac{(\Delta - 3)(\varepsilon - 1)}{2\varepsilon - 1}. \tag{24}$$

Note that, for a fixed ε, $(\Delta-3)(\varepsilon-1)/(2\varepsilon-1) \leqslant (\Delta+1)(\varepsilon-1)/(2\varepsilon-1)$ and that (16) is decreasing with ε, while (24) is increasing. These two expressions asymptotically coincide when

$$\frac{\Delta+1}{\varepsilon}+2 = 2+\frac{(\Delta+1)(\varepsilon-1)}{2\varepsilon-1} \Longleftrightarrow \varepsilon^2-3\varepsilon+1=0 \overset{\varepsilon>1}{\Longrightarrow} \varepsilon_0 \backsimeq 2.62. \tag{25}$$

Setting $\varepsilon_0 = 2.62$, we get $c_{\mathtt{C2\_OLVC}} \leqslant (\Delta+6.24)/2.62$. This, for large values of Δ, is asymptotically equal to $\Delta/2.62$. $\square$

4.3. *Lower bounds for the competitive ratio*

4.3.1. *Case $t > 2$*

The ideas in the proof of Theorem 1 can be used even when the input-graph is revealed in t clusters (with $t = \mathrm{o}(n)$). The main difficulty for such a generalization consists of controlling the growth of the number of vertices (due to Rule ii of Item 1 of Theorem 1) when the number of clusters is fixed (assuming that clusters are non-empty). We exhibit a value of t $(1 \ll t \ll n)$ for which such difficulty can be overcome. In all, we prove the following theorem (the proof of which being quite technical, it is given in appendix).

Theorem 5. *When $t = c\sqrt{n}\log n$, for some constant c, no on-line algorithm for* ON-LINE VERTEX COVER *can achieve competitive ratio smaller than $\Delta - 2$, against an optimal off-line algorithm, even if the input-graph is a tree, any cluster is non-empty and n is known in advance.*

Similar results should be possible for other values of t also. But it seems difficult to produce a global result working for any value of t.

4.3.2. *Limits on the competitiveness for $t = 2$*

As previously in Section 3.3, we present a graph and a strategy for revealing it in two steps such that every on-line VC-algorithm cannot achieve competitive ratio better than the bound provided.

Theorem 6. *For $t = 2$ and for all $\Delta \geqslant 2$, no algorithm can achieve competitive ratio strictly better than $(\Delta+1)/2$ for a graph of maximum degree Δ, even if it is bipartite (denoted by $H(V_1 \cup V_2, E)$) with no isolated vertices, $|V_1| = |V_2|$, V_1 is the first cluster (and V_2 is the second one), both clusters have the same size and a graph G isomorphic of H is known in advance.*

Proof. Given an integer k and two sets $A = \{1, \dots, |A|\}$ and $B = \{1, \dots, |B|\}$ such that $|B| = k|A|$, we set $A \times_k B = \{(a_i b_{(i-1)k+j}) \in A \times B, i \in \{1, \dots, |A|\}, j \in \{1, \dots, k\}\}$. In other words, if A and B are vertex-sets, the graph $(A \cup B, A \times_k B)$ consists of $|A|$ stars of size $k+1$ rooted in the vertices of A.

Let $\Delta \geqslant 2$ be a fixed integer and set

$$n = 2\Delta(\Delta + 1). \tag{26}$$

We define $H(V_1 \cup V_2, E)$ where

- $V_1 = N_1^1 \cup N_1^2$ (sets N_1^1 and N_1^2 are mutually disjoint) with $|N_1^1| = \Delta$, $|N_1^2| = \Delta^2$, $N_1^1 \cap N_1^2 = \emptyset$;
- $V_2 = N_2^1 \cup N_2^2$ (sets N_2^1 and N_2^2 are mutually disjoint) with $|N_2^1| = \Delta^2$, $|N_2^2| = \Delta$, $N_2^1 \cap N_2^2 = \emptyset$;
- $E = (N_1^1 \times_\Delta N_2^1) \cup (N_2^2 \times_\Delta N_1^2)$.

The so-constructed graph H is bipartite, without isolated vertices and a minimum cardinality vertex covering of H is of size $\tau(H) = n/(\Delta + 1) = 2\Delta$ (by (26)).

Consider any on-line algorithm and denote it by OLVC. We will show that Player 1 can reveal H in such a way that OLVC will include at least $\Delta^2 + \Delta$ vertices in the cover (inducing so a competitive ratio $(\Delta + 1)/2$).

First cluster is V_1 (an independent set of size $\Delta(\Delta + 1)$). Let N_1 be the set of vertices of V_1 introduced in the solution by OLVC and set $n_1 = |N_1|$. We consider the two following cases:

1. $n_1 \leqslant \Delta^2$;
2. $n_1 > \Delta^2$.

For Case 1, Player 1 can reveal the second cluster in such a way that $N_1 \subseteq N_1^2$ and $\lfloor n_1/\Delta \rfloor$ vertices of N_2^2 are each one linked with Δ vertices of N_1 (satisfying the shape of H). Then, OLVC necessarily includes N_2^1 in the cover together with $\Delta - \lfloor n_1/\Delta \rfloor$ vertices of N_2^2. So, the constructed solution has size at least $\Delta^2 + n_1 + \Delta - \lfloor n_1/\Delta \rfloor \geqslant \Delta^2 + \Delta$ vertices.

For Case 2, Player 1 reveals the second cluster in such a way that $N_1 = N_1^2 \cup R_1$, where $R_1 \subseteq N_1^1$ (satisfying the shape of H). Then, since OLVC has to take in the solution the $\Delta(\Delta - |R_1|)$ vertices of N_2^1 non-adjacent to vertices of R_1, and $|R_1| \leqslant \Delta$, the constructed cover is of size at least $\Delta^2 + |R_1| + (\Delta - |R_1|)\Delta \geqslant \Delta^2 + \Delta$. □

Note that if we allow isolated vertices in G, one can easily show that one cannot guarantee competitive ratio strictly better than Δ.

5. A model based on-line arrival of edges

Let us note that one can consider other on-line models more or less complicated than the ones just considered. In this section, we consider a simple model assuming that the input-graph is revealed by means of its edges rather than of its vertices. They arrive one at a time and for any new edge, the links of its endpoints with the endpoints of the edges already present are also revealed. We suppose that $|E|$ is known in advance, we set $E = \{e_1, \ldots, e_{|E|}\}$, where e_i are numbered in order of their arrival. For any e_i just arrived, if $X(e_i) \cap C = \emptyset$, then $X(e_i)$ is included in C (the vertex cover under construction).

In this case, the irrevocability in the construction of the on-line solution deals with the endpoints of an edge as a whole. With respect to a model based upon arrival of vertices it is as one allows, for every edge arriving, a kind of backtracking of level one.

Proposition 2. *The competitive ratio of the algorithm against an optimal off-line algorithm is bounded above by* 2 *and that this bound is tight*.

Proof. Assume that the endpoints of q edges have entered C. It is easy to see that these edges form a maximal matching of G (of size q). The set C finally computed by the algorithm verifies $|C| = 2q$. On the other hand, by (1), $\tau(G) \geqslant q$. The competitive bound 2 is then immediately deduced.

For tightness, consider a star revealed edge-by-edge. The algorithm will introduce in C the endpoints of the first edge revealed and no new vertex will be introduced in C later. The optimal vertex-cover for any star consists of its center. So here, the bound 2 is attained. □

The on-line model just described is equivalent to the one where all vertices are present from the beginning of the game and edges are presented one-by-one. Here, whenever an edge arrives none of the endpoints of which are in C, then both of its endpoints enter C.

6. Allowing backtracking

In this section we somewhat change the working hypotheses adopted and suppose that one can go back over the solution constructed during previous steps. We assume that one can change this solution but she/he has to pay some cost for doing it. Let us note that the backtracking model dealt here allows only adding vertices when the whole graph is revealed.

Our on-line algorithm for the case of the backtracking is basically `t_OLVC`. The spirit of our thought process can be outlined as follows. The best approximation ratio known for VC is bounded above by 2 (this ratio is equal to $2 - (\log \log n / \log n)$ [2]). On the other hand, ON-LINE VERTEX COVER being computationally harder, it is a priori worse approximated than VC. So, one can "restrain" her/himself in searching for competitive ratios as near as possible to 2. The maximum matching performed on each cluster of G by `t_OLVC` obviously guarantees approximation ratio 2 on any cluster. The fact that the whole competitive ratio is finally "deteriorated" is due to the vertices of the graph B that have to be taken into account in order to cover cross-edges, i.e. edges between clusters. So the algorithm we propose in what follows starts with running `MAX_MATCHING` on each cluster and by delaying its decision on the cross-edges (in other words, the exposed vertices of any cluster are firstly considered as not belonging to the solution under construction). Next, once all clusters are revealed, graph G_B is formed and `MAX_MATCHING(B)` is run. The final solution is the union of the endpoints of all the edges retained by the successive runs of `MAX_MATCHING`. In what follows, for $V' \subseteq V$, we denote by $G[V']$ the subgraph of G induced by V'.

Let us consider the model where the cost due to the change of the status of x vertices is $f(x)$ where f is some given positive function satisfying $f(0) = 0$. We assume that the cost of the final solution equals the number of the vertices added to the covering during the execution of the on-line computation plus $f(x)$, where x is the number of vertices added to the covering when the whole graph is revealed; we also consider that the optimal cost is $\tau(G)$. It is quite natural to assume that f is continuous and not decreasing. In what follows, we restrict ourselves to the case where the slope of *f* (more precisely the average unit cost $f(x)/x$, $x > 0$) is monotonous. This seems to us to be the most natural case. Similar analyses could be conducted for other kinds of cost-functions, but they should require specific formulations for *f*.

6.1. $f(x) = \kappa x$, *for some* $\kappa > 0$

We assume $\kappa > 1$ (the case $\kappa \leqslant 1$, i.e. the one where postponing the decision is beneficial is not natural in on-line computation) and consider the following algorithm denoted by `Bt_OLVC`:

- for $i = 1, \dots, t$, set $C = X(\texttt{MAX\_MATCHING}(\texttt{G}_\texttt{i}))$;
- set $B = G[\cup_{i=1}^{t}(V_i \setminus X(\texttt{MAX\_MATCHING}(\texttt{G}_\texttt{i}))]$;
- output $C = C \cup X(\texttt{MAX\_MATCHING}(\texttt{B}))$.

Also, as previously, for $i = 1, \dots, t$, we set $M_i = \texttt{MAX\_MATCHING}(\texttt{G}_\texttt{i})$ and denote by m_i the cardinality of M_i; also, we set $M_B = \texttt{MAX\_MATCHING}(\texttt{B})$ and denote by m_B the cardinality of M_B.

Proposition 3. *The competitive ratio of* `Bt_OLVC` *against an optimal off-line algorithm for VC is bounded above by* 2κ.

Proof. In fact, as one can see from `Bt_OLVC`, the vertices changed belong to $\cup_{i=1}^{t}(V_i \setminus X(M_i))$. Among these vertices, exactly $|X(M_B)| = 2m_B$ vertices pass from non-covering to covering ones. Suppose that for each of them a cost κ has to be paid. In this case, the cost of the solution computed by `Bt_OLVC` is smaller than, or equal to, $(2\sum_{i=1}^{t} m_i) + 2\kappa m_B$. Using (4), for the optimal cost, we immediately get: $c_{\texttt{Bt\_OLVC}} \leqslant 2\kappa$. $\square$

Suppose now that we do not require polynomial execution times and consider the following algorithm, denoted by `Ot_OLVC`:

- for $i = 1, \dots, t$, include in the solution an optimal vertex cover of G_i;
- set *B* the graph induced by the uncovered edges;
- complete the solution by an optimal cover of *B*.

Using the facts that $\tau(G) \geqslant \sum_{i=1}^{t} \tau(G_i)$ and that $\tau(G) \geqslant \tau(B)$, by an analysis similar to the one for `Bt_OLVC`, we conclude a competitive ratio of $1 + \kappa$.

6.2. $f(x)/x$ *decreases with x*

A decreasing average unit-cost is a very usual economic model. In this case, the hypotheses on *f* imply that, for any $x > 0$, $f(x) \leqslant \kappa x$, where $\kappa = f(1)$. Therefore, the result of Proposition 3 remains valid and the following corollary holds.

Corollary 4. *For the case where* $f(x)/x$ *decreases with x,* `Bt_OLVC` *achieves competitive ratio* $\max\{2, 2f(1)\}$.

Consider now that $f(1) > 1$ and assume that $\lim_{x\to\infty} f(x)/x < 1$. Then, there exists x_0 such that $f(x_0) = x_0$. Since *f* is supposed fixed, one can assume that x_0 is known. In this case, we can consider the following on-line algorithm for ON-LINE VERTEX COVER, (where we suppose that the size n of the final graph is known in advance):

- if $n \leqslant x_0$, then apply `Ot_OLVC`;
- wait until the whole graph is revealed and apply `MAX_MATCHING` in order to compute a 2-approximation.

If $n \leqslant x_0$, then a competitive ratio of $1 + f(1)$ is polynomially achieved (since x_0 is supposed to be known); otherwise, a competitive ratio 2 is guaranteed (see Section 6.1 dealing with $\kappa \leqslant 1$).

Suppose finally that $\lim_{x\to\infty} f(x)/x = \kappa_1 \geqslant 1$. Then, for any $\varepsilon > 0$, there exists x_ε, such that for any $x \geqslant x_\varepsilon$, $f(x) \leqslant (\kappa_1 + \varepsilon)x$. We then consider the following on-line algorithm for ON-LINE VERTEX COVER (always supposing that the size n of the whole graph is known in advance):

- if $n \leqslant x_\varepsilon$, then apply `Ot_OLVC`;
- apply `Bt_OLVC`.

The algorithm just above obviously guarantees competitive ratio $\max\{1 + f(1), 2(\kappa_1 + \varepsilon)\}$ and the following proposition summarizes the discussion of this section.

Proposition 4. *If $f(x)/x$ decreases with x, then, for any $\varepsilon > 0$, there exists a polynomial time on line algorithm achieving competitive ratio* $\max\{2, 1 + f(1), 2(\kappa_1 + \varepsilon)\}$, *where* $\kappa_1 = \lim_{x\to\infty} f(x)/x$.

6.3. $f(x)/x$ increases with x

By similar arguments as in Section 6.2, one can prove that if $\lim_{x\to\infty} f(x)/x = \kappa_2 < \infty$, then a competitive ratio $\max\{2, 2\kappa_2\}$ can be achieved. So we can now assume $\lim_{x\to\infty} f(x)/x = \infty$.

If $f(1) < 1$, then we distinguish two cases, namely, $n \leqslant x_0$ and $n > x_0$, where, as previously, x_0 is such that $f(x_0) = x_0$ (recall that n denotes the order of the whole graph).

The first case can be faced by the same arguments as in Section 6.2.

Consider now case $n > x_0$ and assume that $f(x)/x > 1$. It can be easily shown that for any n there exists $r_n \in (0, n]$ such that

$$\frac{f(2r_n)}{2r_n} = \frac{n}{r_n} \tag{27}$$

i.e., $f(2r_n) = 2n$. Note that r_n can be polynomially computed by dichotomy.

For the cost-model dealt, we somewhat modify `Bt_OLVC` and assume n known in advance. We so derive the following algorithm denoted by `Mt_OLVC`:

(a) arrival of G_1: set $C = X(\mathtt{MAX\_MATCHING(G_1)})$;
(b) arrival of G_i, $i = 2, \ldots, t$:
- set: $B = G[\cup_{j=1}^{i}(V_j \setminus X(\mathtt{MAX\_MATCHING(G_j)}))]$ and $M_B = \mathtt{MAX\_MATCHING(B)}$;
- if $m_B \leqslant r_n$, then set $C = C \cup X(\mathtt{MAX\_MATCHING(G_i)})$, else set $C = C \cup V_i$;

(c) let i_0 be the last i for which the then instruction of Step (b) is executed; set $B' = G[\cup_{j=1}^{i_0}(V_j \setminus X(\mathtt{MAX\_MATCHING(G_j)}))]$;
(d) output $C = C \cup X(M_{B'})$.

Theorem 7. *If the change of the status of x vertices induces a cost $f(x) > x$, where $f(x)/x$ increases with x, then the competitive ratio of* `Mt_OLVC` *against an optimal off-line VC-algorithm is bounded above by $3n/r_n$, where r_n is such that $f(2r_n) = 2n$.*

Proof. Note first that the only vertex-changes performed by `Mt_OLVC` are on $X(M_{B'})$ (where B' is the graph constructed in Step (c)) and, furthermore, that $m_{B'}$ always satisfies $m_{B'} \leqslant r_n$ and, consequently

$$\frac{f(2m_{B'})}{2m_{B'}} \leqslant \frac{f(2r_n)}{2r_n} \overset{(27)}{=} \frac{n}{r_n},$$
$$f(2m_{B'}) \leqslant 2m_{B'}\frac{n}{r_n}. \tag{28}$$

If the else statement of Step (c) in `Mt_OLVC` is not executed at all, i.e., if $i_0 = t$ (Step (c)), then $m_B \leqslant r_n$, where, obviously, $B = B' = G[\cup_{j=1}^{t}(V_j \setminus X(\mathtt{MAX\_MATCHING}(\mathtt{G_j})))]$. Consequently, using (28)

$$c_{\mathtt{Mt\_OLVC}} \leqslant \frac{2\sum_{i=1}^{t} m_i + f(2m_{B'})}{\sum_{i=1}^{t} m_i + m_{B'}} \leqslant \max\left\{2, \frac{f(2m_{B'})}{m_{B'}}\right\}. \tag{29}$$

Suppose, without loss of generality, that the max in (29) is realized by the term $f(2m_{B'})/m_{B'}$; so, (29) becomes

$$c_{\mathtt{Mt\_OLVC}} \leqslant \frac{f(2m_{B'})}{m_{B'}} = 2\frac{f(2m_{B'})}{2m_{B'}} \leqslant \frac{2f(2r_n)}{2r_n} \overset{(27)}{=} \frac{2n}{r_n}. \tag{30}$$

On the other hand, suppose that the else statement of Step (b) is executed at least once. Then

$$m_B > r_n. \tag{31}$$

Using (28) and denoting by $v(C)$ the value of the set C computed by `Mt_OLVC`, we get

$$v(C) \leqslant n - 2m_{B'} + f(2m_{B'}) \overset{(28)}{\leqslant} n + 2m_{B'}\frac{n}{r_n}. \tag{32}$$

Denote by m the cardinality of a maximum matching of G and use (4), (31) and (32). Then, $c_{\mathtt{Mt\_OLVC}} = v(C)/\tau(G) \leqslant (n/r_n) + 2(n/r_n) = 3n/r_n$, that concludes the proof of the theorem. $\square$

Corollary 5. *Assume that $f(x)/x$ increases with x and that $x < f(x) \leqslant x^p$, for any $p \in \mathbb{N}$. Then, the competitive ratio of* `Mt_OLVC` *against an optimal off-line VC-algorithm is bounded above by $6n^{1-1/p}$.*

Indeed, f satisfies the conditions of Theorem 7. Furthermore, the hypotheses on f imply that $r_n \geqslant n^{1/p}/2$. Therefore, the result claimed in Corollary 5 follows immediately.

We now show that the result of Theorem 7 is quite tight, since no on-line algorithm can achieve competitive ratio (against an optimal off-line one) better than cr_n for some constant c.

Proposition 5. *The result of Theorem 7 is asymptotically non-improvable. Indeed, there exist cost-functions verifying the hypotheses of Theorem 7 such that no on-line algorithm can achieve, against an optimal off-line algorithm, competitive ratio cn/r_n for any $c < 1/2$, even if $t = 2$ and the final graph is bipartite.*

Proof. Assume cost-function $f : x \mapsto x^2$ (it clearly verifies the hypotheses of Theorem 7), let $(\Delta, n_1) \in \mathbb{N} \times \mathbb{N}$ and set $n = (1 + \Delta)n_1$. At the first step, V_1 is an independent set of size n_1. If Player 2 chooses some vertices of V_1, then the whole instance is a graph without any edge. In this case, the optimal value $\tau^*(G)$ is 0, whereas the on-line value is positive. The resulting ratio equals ∞ and the theorem holds.

Consequently, we can focus ourselves to the case where Player 2 does not choose any vertex during the first step. In this case the graph consists of n_1 stars of size $(1 + \Delta)$ rooted in V_1, one star per vertex in V_1. Then, optimal value satisfies (recall that $n = (1 + \Delta)n_1$)

$$\tau^*(G) = \frac{n}{\Delta + 1} = n_1. \tag{33}$$

Denote by V_1' the set of vertices of V_1 that are changed in order to be included in the final solution (i.e., the vertices introduced in the solution after the backtracking). Then, the solution C computed by Player 2 verifies $C \supseteq V_1' \cup V_2 \setminus \Gamma(V_1')$ for a total cost of

$$v(C) \geqslant f(|V_1'|) + \Delta(n_1 - |V_1'|). \tag{34}$$

Consequently, Player 2 chooses, at best, a set V_1' of cardinality

$$\beta^* \in \operatorname*{Argmin}_{\beta \in \left[0, \frac{n}{\Delta+1}\right]} \left\{\beta^2 - \Delta\beta + \frac{n\Delta}{\Delta + 1}\right\}. \tag{35}$$

Let c and $\varepsilon > 0$ be such that $1/\sqrt{2(2+\varepsilon)} > c$ (clearly, $c < 1/2$). Define then $n_1 = 1/\varepsilon$ and $\Delta = 2n_1$. One can easily show that (35) implies $\beta^* = n_1$; by (34), $v(C) = n_1^2$. Using it together with (33) and taking into account that $\beta^* = n_1$, we get: $v(C)/\tau^*(G) = n_1 = r_n/\sqrt{2(2+\varepsilon)} > cn/r_n$. $\square$

7. Conclusions

On-line computation is actually a very active area of the theoretical computer science. It is a domain of great interest for operational researchers also, from both theoretical and practical points of view, since the mathematical problems here emanate from models expressing reality more richly than the conventional ones. The vertex-covering problem dealt in this paper is one of the central problems in combinatorial optimization in its off-line version. As we have seen at the beginning of the paper, it remains very natural even in its on-line version. There exists a number of open problems that seem interesting for further studies. First of all, the improvement, if possible, of the competitive ratios obtained and the achievement of lower bounds for the case where $t < n$. Also, a further generalization of the vertex-covering is the one where we consider weights on the vertices of the input-graph and we search for a minimum total-weight vertex cover. In the company model presented in Section 1, this generalization has a very natural interpretation if we consider that the manufacturing of an order has its proper cost and the company wishes to minimize the cost of the manufacturing in subcontracting. Performing a competitive analysis of on-line algorithms for this weighted version of ON-LINE VERTEX COVER seems to us a very interesting open problem.

Acknowledgements

The very pertinent and useful remarks, comments and suggestions of two anonymous referees are gratefully acknowledged.

Appendix A. Proof of Theorem 5

The main idea of the proof is analogous to the one of the proof of Item 2 in Theorem 1. The extra difficulty here is because of $t = o(n)$. It should be noted that the fact $t = n$ in Theorem 1 importantly simplifies the respective proof.

As in Section 3.3, we consider a two-player game. Player 1 reveals the instance by clusters while Player 2 constructs the solution, i.e., it partitions V_i into two subsets: S_i and C_i, the former one denoting the set of the independent vertices of V_i and the latter one denoting the set of the covering vertices of V_i. We denote by `PLAY2` the procedure, representing construction of the decision of Player 2, about the partition of V_i. In other words, the decision of Player 2 will be denoted by $(C_i, S_i) =$ `PLAY2(Vi)`. Moreover, we call *badly covered* a *d*-leaves star whose all leaves are included in the vertex cover under construction. The revealing strategy of Player 1 is based upon the following module denoted by `ONE_VERTEX` and dealing with the revealing of a single vertex. It is called with inputs $\Delta \in \mathbb{N}$, $G[\cup_{j=1,\dots,i} V_i]$, i.e., the graph already revealed, and two sets of vertices R_S and R_C; it returns a new vertex y, its links with the vertices already revealed and the sets R_S and R_C updated. The neighbors of x considered in the third "if" of algorithm `ONE_VERTEX` deal with the graph already revealed. For reasons of simplicity, we set in what follows $G[\cup_{j=1,\dots,i} \ V_i] = [G]_i$. Module `ONE_VERTEX` works as follows:

1. if $|R_C| \geqslant \Delta$, then y is connected to Δ vertices of R_C; these vertices are removed from R_C;
2. if $|R_C| < \Delta$, then
   (a) if $R_S \neq \emptyset$, then: choose $x \in R_S$; y is linked only with x; if x has Δ neighbors, then set $R_S = R_S \setminus \{x\}$;
   (b) if $R_S = \emptyset$, then y is isolated.

Remark that a star is built by Step 1 and by the second if of Step (2a). Moreover, in the overall revealing algorithm, we also use the following additional module `UPDATE(X1, Y1, X2, Y2)`: *if X_1 contains an isolated vertex, then add it in X_2; if Y_1 contains an isolated vertex, add it in Y_2*. Assume $\Delta \geqslant 16$ be an integer. Define $K' = \lceil 1 + 2\log \Delta \rceil$ and $K = \Delta - 2K' - 2 \geqslant 0$. The final graph G has a form analogous to the one of Fig. 1. It consists of

- $\Delta - 1$ stars of size $\Delta + 1$ plus a tree of size $\Delta + 1$ with a vertex cover of size 2;
- a root-vertex such that the whole graph is a tree of degree Δ.

Assume $t = \lceil 2\log \Delta \rceil + (2 + K')(1 + \Delta) + 1$ and consider the following strategy played by Player 1 for revealing the graph in t steps; this strategy is called `GAME` in what follows:

- set: $K' = \lceil 1 + 2\log \Delta \rceil$, $K = \Delta - 2K' - 2$, $S = \emptyset$, $C = \emptyset$, $i = 0$;
- `PHASIS` 1: main phasis
  (1.1) set $A = 0$;
  (1.2) while $A < K\Delta$: set $i = i + 1$; let V_i be an independent set of size $\lceil K - A/\Delta \rceil$; set: $(C_i, S_i) = \mathtt{PLAY2}(\mathtt{V_i})$, $S = S \cup S_i$, $C = C \cup C_i$, $A = A + \Delta|S_i| + |C_i|$;
  (1.3) set $r = A - K\Delta$; let R_C^1 be a subset of C of cardinality r; set: $X_S = S$, $X_C = C \setminus R_C^1$;
- `PHASIS` 2: adjustment of the number t of steps
  set $R_S^2 = \emptyset$, $t = \lceil 2\Delta \log \Delta \rceil - i$, $R_C^2 = R_C^1$, $H_2 = R_C^2$; for $j = 1, \dots, t$: run `ONE_VERTEX` and set: $y = \mathtt{ONE\_VERTEX}(\Delta, [\mathtt{G}]_\mathtt{i}, \mathtt{R_S^2}, \mathtt{R_C^2})$; set: $i = i + 1$, $V_i = \{y\}$, $H_2 = H_2 \cup \{y\}$; set: $(C_i, S_i) = \mathtt{PLAY2}(\mathtt{V_i})$ (in this case $|V_i| = 1$); set: $S = S \cup S_i$, $C = C \cup C_i$; $\mathtt{UPDATE}(\mathtt{S_i}, \mathtt{C_i}, \mathtt{R_S^2}, \mathtt{R_C^2})$;
- `PHASIS` 3: re-adjustment of the number of blocks
  (3.1) set: $R_C^3 = \emptyset$, $R_S^3 = \emptyset$, $H_3 = \emptyset$;
  (3.2) for $j = 1, \dots, K'(1 + \Delta)$:
  - if $|H_2| < K'(\Delta + 1)$, then set $\ell = 2$; else set: $\ell = 3$, $V_i^2 = \emptyset$;
  - for $k = \ell$ to 3: set: $y = \mathtt{ONE\_VERTEX}(\Delta, [\mathtt{G}]_\mathtt{i}, \mathtt{R_S^k}, \mathtt{R_C^k})$, $V_i^k = \{y\}$, $H_k = H_k \cup V_i^k$;
  - set: $V_i = V_i^2 \cup V_i^3$, $i = i + 1$, $(C_i, S_i) = \mathtt{PLAY2}(\mathtt{V_i})$ (here $|V_i| \leqslant 2$), $S = S \cup S_i$, $C = C \cup C_i$;
  - for $k = \ell$ to 3: $\mathtt{UPDATE}(\mathtt{S_i} \cap \mathtt{V_i^k}, \mathtt{C_i} \cap \mathtt{V_i^k}, \mathtt{R_S^k}, \mathtt{R_C^k})$;
  - for $i = 2, 3$, set u_i the degree of the vertex of R_S^i (if non-empty; remark that $|R_S^i| \leqslant 1$);
- `PHASIS` 4: last two blocks (steps (4.1) and (4.2) construct the last star, while steps (4.3) and (4.4) construct the last block)
  (4.1) set $i_4 = 0$, $R_C^4 = R_C^2 \cup R_C^3$ ($|R_C^4| \leqslant 2\Delta - 2$), $R_S^4 = \emptyset$;
  (4.2) repeat until a star is built by `ONE_VERTEX`: set: $i = i + 1$, $i_4 = i_4 + 1$; set: $y = \mathtt{ONE\_VERTEX}(\Delta, [\mathtt{G}]_\mathtt{i}, \mathtt{R_S^4}, \mathtt{R_C^4})$, $V_i = \{y\}$, $(C_i, S_i) = \mathtt{PLAY2}(\mathtt{V_i})$, $C = C \cup C_i$, $S = S \cup S_i$; $\mathtt{UPDATE}(\mathtt{S_i}, \mathtt{C_i}, \mathtt{R_S^4}, \mathtt{R_C^4})$
  (4.3) (at the beginning of the step, $R_S^4 = \emptyset$)) repeat until $R_S^4 \neq \emptyset$ or $|R_C^4| = \Delta$: set: $i = i + 1$, $i_4 = i_4 + 1$; set: $y = \mathtt{ONE\_VERTEX}(\Delta, [\mathtt{G}]_\mathtt{i}, \mathtt{R_S^4}, \mathtt{R_C^4})$, $V_i = \{y\}$, $(C_i, S_i) = \mathtt{PLAY2}(\mathtt{V_i})$, $C = C \cup C_i$, $S = S \cup S_i$; $\mathtt{UPDATE}(\mathtt{S_i}, \mathtt{C_i}, \mathtt{R_S^4}, \mathtt{R_C^4})$;
  (4.4) if $R_S^4 = \emptyset$, then a new vertex y, linked to any vertex of R_C^4, is revealed; set: $V_i = \{y\}$, $(C_i, S_i) = \mathtt{PLAY2}(\mathtt{V_i})$, $i = i + 1$, $i_4 = i_4 + 1$, $S = S \cup S_i$, $C = C \cup C_i$; else ($|R_S^4| = 1$) $\Delta - |R_C^4|$ vertices are revealed; let V_i be their set; they are linked to the (unique) vertex of R_S^4 and one of them is linked to any vertex of R_C^4 (if non-empty); $(C_i, S_i) = \mathtt{PLAY2}(\mathtt{V_i})$, $i = i + 1$, $i_4 = i_4 + 1$, $S = S \cup S_i$, $C = C \cup C_i$;
- `PHASIS` 5: completion of G by the last clusters; the set H_5 of vertices to be revealed during this phasis consists of
  - for any vertex $x \in X_S$, Δ independent vertices linked to x;
  - $|X_C|/\Delta$ vertices each one linked to Δ proper independent vertices of X_C;
  - $\Delta - u_i$ vertices linked to R_S^i, $i = 2, 3$;
  - the root-vertex r;

(5.1) set $i_5 = 2(\Delta + 1) - i_4 + 1$;
(5.2) partition H_5 into i_5 non-empty clusters V_j, $j = 1, \ldots, i_5$ such that r belongs to the last one;
(5.3) for $j = 1$ to i_5: set $(S_j, C_j) = \mathtt{2\_PLAY(V_j)}$, $C = C \cup C_j$, $S = S \cup S_j$.

Lemma 2. 1. *`PHASIS 1` of `GAME` takes at most* $\lceil 2\Delta \log \Delta \rceil$ *steps and at the end*, $K\Delta \leqslant A < (K+1)\Delta$, $r < \Delta$, $|X_C|/\Delta \in \mathbb{N}$; *moreover*, $|X_S| + |X_C|/\Delta = K$.
2. *At the end of `PHASIS 2`*, $i = \lceil 2\Delta \log \Delta \rceil$ *and* $|H_2| \leqslant K'(\Delta + 1)$.

Proof of Item 1. From the size of the independent set V_i computed in Step (1.2), one gets $\Delta|S_i| + |C_i| \leqslant \Delta|V_i| < (K+1)\Delta - A$. Hence, the current value of A satisfies $A < (K+1)\Delta$, so $r < \Delta$. On the other hand, clearly, $A = \Delta|S| + |C|$. Consequently, at the end of Step (1.2), one has $|C| = q\Delta + r$ with $q + |S| = K$. So, $|X_C| = q\Delta$.

We now show that Step (1.2) is executed at most $\lceil 2\Delta \log \Delta \rceil$ times. Denote by A_i the value of A at the end of the ith execution of the loop. Sequence $(A_i)_i$ satisfies, $\forall i$, such that $A_i < K\Delta$

$$A_{i+1} \geqslant A_i + K - \frac{A_i}{\Delta}. \tag{A.1}$$

Let now $B_i = K\Delta - A_i$. Sequence $(B_i)_i$ satisfies (using (A.1)) the following induction rule:

$$B_0 = K\Delta, \tag{A.2}$$

$$B_{i+1} \leqslant B_i \left(1 - \frac{1}{\Delta}\right) \forall i \text{ such that } B_i > 0. \tag{A.3}$$

From (A.2), one can deduce that, $\forall i$, $B_{i-1} > 0$ and $B_i \leqslant K\Delta(1 - \Delta^{-1})^i$. Furthermore, from (A.1) and (A.2), B_i becomes non-positive for $i > \log(K\Delta) / -\log((1 - \Delta^{-1})^i)$. This last quantity is smaller than $2\Delta \log \Delta$. In fact, $\log(K\Delta)/(-\log(1 - \Delta^{-1})) \leqslant \Delta \log (K\Delta) \leqslant 2\Delta \log \Delta$. Consequently, Step (2) is not executed more than $\lceil 2\Delta \log \Delta \rceil$ times and the proof of Item 1 is complete.

Proof of Item 2. The value of i claimed follows immediately from the total number of the iterations during `PHASIS 2`. Furthermore, $|H_2| = |R_C^1| + t \leqslant \Delta + \lceil 2\Delta \log \Delta \rceil \leqslant (\Delta + 1)(\lceil 2 \log \Delta \rceil + 1)$. $\square$

From algorithm `ONE_VERTEX` we can deduce that at the end of `PHASIS 2`, the graph revealed consists of $|X_S|$ isolated independent vertices, of $|X_C|$ $(= \Delta(K - |X_S|))$ isolated covering vertices, of s_2 badly covered stars of size $\Delta + 1$, of σ_2 badly covered stars of size $u_2 + 1$, $u_2 < \Delta$ and of $v_2 < \Delta$ isolated covering vertices. Moreover, $\sigma_2 = |R_2^S| \in \{0, 1\}$ and $v_2 = |R_2^C| \leqslant \Delta$.

Lemma 3. *At the end of `PHASIS 3`*, $|H_2| = |H_3| = (\Delta + 1)K'$, $i = \lceil 2\Delta \log \Delta \rceil + (\Delta + 1)K'$.

Proof. It follows immediately from the fact that, at the beginning of PHASIS 3, $|H_2| \leqslant (\Delta + 1)K'$. □

At the end of PHASIS 3 the graph revealed consists of $|X_S|$ isolated independent vertices, of $|X_C|$ isolated covering vertices, $|X_S| + |X_C|/\Delta = K$, and of the subgraphs of the final graph induced by H_2 and H_3 created during PHASES 2 and 3, i.e., of $s_2 + s_3$ badly-covered stars of size $\Delta + 1$, of $|R_S^2| + |R_S^3|$ badly-covered stars of sizes $u_2 + 1$ and $u_3 + 1$, respectively, $u_2, u_3 \leqslant \Delta$, and of $|R_C^2| + |R_C^3|$ isolated vertices introduced in the solution, $|R_C^i| \leqslant \Delta$, $i = 2, 3$; moreover, $|R_S^i| \leqslant 1$, $(u_i + 1)|R_S^i| + |R_C^i| = (\Delta + 1)|R_S^i|$ and $s_i + |R_S^i| = K'$.

Lemma 4. *At the end of PHASIS 4, the following holds*:
1. $2 \leqslant i_4 \leqslant 2(\Delta + 1)$ and $i = \lceil 2 \log \Delta \rceil + (\Delta + 1)K' + i_4$;
2. *the graph already revealed consists of* $|X_S| + |X_C|$ *isolated vertices such that* $|X_S| + |X_C|/\Delta = K$, *of* $s_2 + s_3$ *badly-covered stars of size* $\Delta + 1$, *of* $|R_S^2| + |R_S^3|$ *badly-covered stars of sizes* $u_2 + 1$ *and* $u_3 + 1$, *respectively*, $u_2, u_3 \leqslant \Delta$, *of one badly covered star of size* $\Delta + 1$ *and of the last bloc, a* 2*-level tree the edges of which are covered by* Δ *vertices*;
3. *it remains* $\Delta|X_S| + |X_C|/\Delta + \Delta - u_2 + \Delta - u_3 + 1$ *vertices to be revealed, with* $\Delta|X_S| + |X_C|/\Delta \geqslant K$.

Proof. The proof is immediate from the algorithm GAME. □

From Item 1 of Lemma 4, $1 \leqslant i_5 \leqslant 2\Delta + 1$. On the other hand, $k + 1 \geqslant 2\Delta + 1$ vertices remain to be revealed (Item 2 of Lemma 2). So, the partitioning in Step (5.2) of PHASIS 5 is indeed possible.

The overall graph has been revealed within $t = O(\Delta \log \Delta)$ steps and is of order $n = O(\Delta^2)$. Consequently, $t = O(\sqrt{n} \log n)$. Furthermore, the final vertex cover constructed by Player 2 has size $|C| \geqslant \Delta^2$, while a minimum vertex cover of the graph is of size $\Delta + 2$. This concludes the proof.

References

[1] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, M. Talamo, Algorithms for the on-line traveling salesman problem, Algorithmica 29 (4) (2001) 560–581.

[2] R. Bar-Yehuda, S. Even, A local-ratio theorem for approximating the weighted vertex cover problem, Ann. Discrete Appl. Math. 25 (1985) 27–46.

[3] C. Berge, Graphs and Hypergraphs, North-Holland, Amsterdam, 1973.

[4] M. Demange, X. Paradon, V. T. Paschos, On-line maximum-order induced hereditary subgraph problems, in: V. Hlaváč, K.G. Jeffery, J. Wiedermann (Eds.), SOFSEM 2000—Theory and Practice of Informatics, Lecture Notes in Computer Science, Vol. 1963, Springer, Berlin, 2000, pp. 326–334.

[5] R. El-Yaniv, A. Fiat, R.E. Karp, G.Turpin, Competitive analysis of financial games, in: Proc. FOCS'92, 1992, pp. 327–333.

[6] A. Fiat, G. Wœginger (Eds.), Online Algorithms: The State of the Art, Lecture Notes in Computer Science, Vol. 1442, Springer, Berlin, 1998.

[7] A. Gyárfás, J. Lehel, Online and first-fit colorings of graphs, J. Graph Theory 12 (1988) 217–227.

[8] M.M. Halldórsson, K. Iwama, S. Miyazaki, S. Taketomi, Online independent sets, Theoret. Comput. Sci. 289 (2) (2002) 953–962.

[9] M.M. Halldórsson, M. Szegedy, Lower bounds for on-line graph coloring, Theoret. Comput. Sci. 130 (1) (1994) 163–174.
[10] S. Irani, A.R. Karlin, Online computation, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, PWS Publishing Company, MA, 1997, pp. 521–565 (Chapter 13).
[11] L. Lovász, M. Saks, W. Trotter, An on-line graph coloring algorithm with sublinear performance ratio, Discrete Math. 75 (1–3) (1989) 319–325.
[12] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[13] D. Sleator, R.E. Tarjan, Amortized efficiency of list update and paging rules, Commun. ACM 28 (2) (1985) 202–208.