Algorithmic Operations Research Vol.4 (2009) 36–48

Greedy Algorithms for On-Line Set-Covering

Giorgio Ausiello

Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma "La Sapienza'

Nicolas Bourgeois Telis Giannakos Vangelis Th. Paschos

LAMSADE, CNRS UMR 7024 and Université Paris-Dauphine

Abstract

We study on-line models for the set-covering problem in which items from a ground set arrive one by one and with any such item c, the list of names of sets that contain it in the final instance is also presented possibly together with some information regarding the content of such sets. A decision maker has to select which set, among the sets containing c, has to be put in the solution in order to cover the item. Such decision has to be taken before a new item arrives and is irrevocable. The problem consists in minimizing the number of chosen sets. We first analyze some simple heuristics for the model in which only names of sets are provided. Then we show non-trivial matching upper and lower bounds for the competitive ratio in the model in which for any item that arrives the content of all sets containing it is also revealed.

Key words: on-line algorithm, greedy algorithm, set-covering, competitive ratio

1. Introduction

Suppose the manager of a supermarket decides, from time to time, to add new products to her catalogue. Each new product can be supplied (at the same price) by a list of possible suppliers and the manager has to decide which supplier to choose without an a priori knowledge of what other products she might wish to sell in the future. For making her own life easier, the manager aims at minimizing the number of suppliers. We call this problem *on-line set-covering*. In more abstract terms, items from a ground set are presented to a decision maker, each one with a list of names of sets that contain it in the final instance, and the decision maker has to select one set to cover the given item, before a new item is presented. The problem consists in minimizing the number of chosen sets and, as it is common in on-line algorithms, the quality of the solution is measured in terms of competitive ratio, that is the ratio between the number of sets chosen by the algorithm and the minimum number of sets needed to cover all items. It has to be noted that in this model the decision maker does not even know the entire ground set in advance.

A similar situation would arise if the editor-in-chief of a new journal would like to cover new topics in her journal and for every new topic she is presented with a list of possible area editors that can cover such topic (among others). In the long term she would like to minimize the number of area editors, without a priori knowing which other topics she might wish to cover in the future.

In this paper we address the on-line set-covering problem and we analyze various on-line greedy solution strategies under different models. For each strategy we provide lower and upper bounds on the competitive ratio that may be achieved.

2. Preliminaries and Related Work

Formally, the *set-covering problem* can be defined as follows: let C be a ground set of n elements and $\mathcal{S}$ a family of m subsets of C such that $\cup_{S\in\mathcal{S}} S = C$; the problem consists of finding a family $\mathcal{S}' \subseteq \mathcal{S}$, of minimum cardinality, such that $\cup_{S\in\mathcal{S}'} S = C$.

The set-covering problem has been extensively studied over the past decades. It has been shown to be **NP**-hard in Karp's seminal paper ([5]) and $O(\log n)$-approximable for both weighted and unweighted cases (see [2], for the former and [4,6,9], for the latter; see also [7] for a comprehensive survey on the subject).

Email: Giorgio Ausiello [ausiello@dis.uniroma1.it], Nicolas Bourgeois [bourgeois@lamsade.dauphine.fr], Telis Giannakos [giannako@lamsade.dauphine.fr], Vangelis Th. Paschos [paschos@lamsade.dauphine.fr].

This approximation ratio is the best achievable, unless **P** $=$ **NP** ([8]).

In the *on-line* version of the set-covering problem, one assumes that the instance is not known in advance but it is provided step-by-step. At each step, an item in the ground set is revealed and, together with it, the names of the sets that contain it in the final instance are revealed. Upon arrival of a new item c, an algorithm (called *on-line algorithm*) has to decide irrevocably which of the sets containing c are to be included in the solution under construction, aiming at minimizing the overall number of sets that will be present in the final solution. The fact that the instance is not known in advance, gives rise to various models specified by the ways in which the final instance is revealed and/or by the amount of auxiliary information that is provided to the on-line algorithm at each step. The objective of an on-line algorithm is to construct a feasible solution whose cardinality is as close as possible to the cardinality of an optimum (off-line) solution. More precisely, as it is customary in on-line algorithms [10], in order to evaluate how "close" the solution obtained by the algorithm is to the optimum solution, we make use of the so-called *competitive ratio* $m(x, y)/\text{opt}(x)$, where $m(x, y)$ is the value of the solution y determined by the on-line algorithm on instance x and $\text{opt}(x)$ is the value of an optimum off-line solution on the same instance.

In this paper we study two basic models for on-line set-covering. Basic underlying hypothesis for both of them is that elements of the ground set arrive one-by-one and have to be processed immediately. This basically means that either they will be immediately covered or they will be left uncovered with the risk that the final solution will be unfeasible. Here we suppose that any newly revealed item is immediately covered at this step unless it has been covered before. In both models, the ground set is not known in advance.

In the first model, called $\mathcal{M}_1$ in the sequel, we assume that, together with an element σ_i, only the names of all the sets containing it in the final instance arrive. In the second model, called $\mathcal{M}_2$ in the sequel, we assume that together with the names of the sets covering σ_i also their content is revealed.

We first address model $\mathcal{M}_1$ and we provide some simple results that show that in this model all heuristics behave very poorly. In particular we first show that if no information other than the names of the sets covering a newly arrived element is given, then the competitive ratio of any deterministic algorithm for this model is $\Omega(n)$. Next, we consider algorithm `TAKE-ALL` that at each step includes in the solution all the sets containing the element just revealed, if such element is still uncovered. We show that this algorithm has tight competitive ratio f, where f is the maximum number of sets in $\mathcal{S}$ that contain a ground element (note that f can even grow exponentially with n). Subsequently, we show that, in model $\mathcal{M}_1$, also randomized algorithms behave rather poorly. The algorithm we consider, called `TAKE-AT-RANDOM`, is a randomized algorithm that at each step i picks a set at random among the ones whose names are revealed and includes it in the solution, if the element revealed at step i is still uncovered. For this algorithm, we show that its expected competitive ratio is bounded below by $\Omega(n^{1-\epsilon})$, for every $\epsilon > 0$.

The main results of the paper concern the study of model $\mathcal{M}_2$. We show that, for this model, the competitive ratio of any algorithm that, for any arriving uncovered element, includes at least one set containing it in the cover, is bounded below by $\sqrt{2n(k^*-1)}/k^*$, where k^* is the cardinality of an optimum off-line cover. We then consider two greedy algorithms. The first one, called `TAKE-LARGEST`, at any step takes one of the largest sets that cover the current item σ_i. We show that the competitive ratio of this algorithm is $O(n)$. We next consider a second algorithm, called `TAKE-LARGEST-ON-FUTURE-ITEMS`, that includes in the solution a set which covers most of the yet uncovered items and we show that such algorithm achieves a competitive ratio of $\sqrt{2n(k^*-1)}/k^*$; hence, it is optimum for model $\mathcal{M}_2$. This algorithm can be seen as the on-line counterpart of the natural greedy (off-line) set-covering algorithm [1]. Hence, analysis of competitiveness of `TAKE-LARGEST-ON-FUTURE-ITEMS` is interesting by its own. As it will be seen later, this algorithm is also interesting even in an off-line setting.

Other versions of the set covering problem in a dynamic or on-line setting have been considered in the literature. In [1], the following on-line model has been studied. An instance $(\mathcal{S}, C)$ of the set-covering problem is supposed to be known in advance, but only a part of it, i.e., a sub-instance $(\mathcal{S}_p, C_p)$ of $(\mathcal{S}, C)$ is gradually revealed over time; this sub-instance is not known in advance. A picturesque way to apprehend the model is to think of the elements of C as lights initially switched off. Elements switch on (get activated) one-by-one. Any time an element c gets activated, the algorithm has to

[1] That progressively includes in the solution one of the sets which cover the most of the still uncovered elements ([4]).

decide which among the sets of $\mathcal{S}$ containing c has to be included in the solution under construction (since we assume that $(\mathcal{S}, C)$ is known in advance, all these sets are also known). In other words, the algorithm has to keep an on-line cover for the activated elements. The algorithm proposed for this model achieves competitive ratio $O(\log n \log m)$ (even if less than n elements of C will be finally switched on and less than m subsets of $\mathcal{S}$ include these elements). The on-line models addressed in our paper differ from the one in [1] mainly by the fact that nothing is known initially. In particular, in [1] a critical use is made of the information about the total number n of items and the total number m of sets in the ground instance, both in the randomized and in the deterministic algorithms.

Other related work can be found in [3] where an approach that could be considered to be at midway between on-line and reoptimization approaches is developed. In this paper, the problem to maintain an approximate set-covering while the instance undergoes limited changes is tackled. In particular, it is shown that if solution $\mathcal{S}'$ has been produced by the natural greedy algorithm achieving approximation ratio $O(\log n)$ ([2]), then after $O(\log n)$ insertions of new elements the initial solution $\mathcal{S}'$ still guarantees the same approximation ratio. In the same spirit lies also the dynamic set-covering model addressed in [11].

The remainder of this paper is organized as follows. Section 3. is devoted to the study of model $\mathcal{M}_1$ for set-covering, while in Section 4. we address model $\mathcal{M}_2$ and provide the main results. Finally, in Section 5., some conclusive considerations are reported.

3. Simple Bounds for Model $\mathcal{M}_1$

In this section, we consider the following model. Assume an arrival sequence $\Sigma = (\sigma_1, \ldots, \sigma_n)$ of the elements of C; the objective is to find, for any $i \in \{1, \ldots, n\}$, a family $\mathcal{S}'_i \subseteq \mathcal{S}$ such that $\{\sigma_1, \ldots, \sigma_i\} \subseteq \cup_{S \in \mathcal{S}'_i} S$ (obviously, for all i, $\mathcal{S}'_i \subseteq \mathcal{S}'_{i+1}$). Once an element σ_i, $i = 1, \ldots,$ is revealed, only the names of the subsets of the whole instance containing σ_i are revealed. In what follows, let $F_i = \{S \in \mathcal{S} : \sigma_i \in S\}$ and $f = \max_{\sigma_i \in C}\{|F_i|\}$. We have called this model $\mathcal{M}_1$.

In this model we prove that any algorithm that chooses the sets in F_i to cover the newly revealed element σ_i in a deterministic way behaves rather badly. Besides, we show that also algorithms that make a randomized choice have a somewhat similar bad behavior.

Proposition 1 *Any deterministic on-line algorithm that only receives the names of the sets containing the arriving ground elements is $\Omega(n)$-competitive.*

Proof. The adversary reveals a first uncovered element along with the names $S_1, \ldots, S_N$ of N sets covering it. He then keeps revealing uncovered elements along with all sets from $S_1, \ldots, S_N$ not already included into the cover, until the algorithm has included all N sets into the cover.

Suppose w.l.o.g. that k ground elements have been presented to the algorithm, each one uncovered by the time of its arrival. With the arrival of the first element, the algorithm has included $S_1, \ldots, S_{l_1}$ in the cover, with the arrival of the second it has included $S_{l_1+1}, \ldots, S_{l_2}$, and so on, until the arrival of the kth element (here the algorithm has included $S_{l_{k-1}+1}, \ldots, S_{l_k} = S_N$ in the cover).

The instance might have been the following: $\mathcal{S} = \{S_1, \ldots, S_N\}$; $C = \{e_1, \ldots, e_{\log N}, 1, \ldots, k\}$, i.e., there exist $n = \log N + k$ ground elements ($k \leqslant N$); the set S_N is the ground set itself; any other set S_i included at step j in the solution is the union of the set $\{1, \ldots, j\}$ with the set of the elements e_p, with p ranging over all the places where the binary expression of i has a 1; finally, the arrival sequence is $1, \ldots, k$. In such a setting, the competitive ratio of the algorithm would be N, i.e., $\Omega(n)$.

Figure 1 illustrates the construction just described. First, element 1 along with set names $S_1, S_2, S_3, S_4, S_5, S_6, S_7$ is presented to the algorithm, who has included S_1, S_2, S_3 into the cover (Figure 1(a)). Next, element 2 along with set names S_4, S_5, S_6, S_7 is presented and the algorithm covers it by including S_4, S_5 (Figure 1(b)). Finally, element 3 is presented along with set names S_6, S_7 and the algorithm has included S_6, S_7 in the cover (Figure 1(c)). In this case, the ground element set might have been $\{e_1, e_2, e_3, 1, 2, 3\}$, the same as S_7, while $S_1 = S_{001} = \{e_1, 1\}$, $S_2 = S_{010} = \{e_2, 1\}$, $S_3 = S_{011} = \{e_1, e_2, 1\}$, $S_4 = S_{100} = \{e_3, 1, 2\}$, $S_5 = S_{101} = \{e_1, e_3, 1, 2\}$ and $S_6 = S_{110} = \{e_2, e_3, 1, 2, 3\}$. ■

It can be immediately seen that, under the given model, any deterministic algorithm that includes a specific set containing a new (uncovered) element σ_i (for example, the set of F_i that comes first in lexicographic order) achieves competitive ratio $O(n)$. Indeed, it chooses at most n sets for an optimum greater than, or equal to 1.

Let us now consider the following algorithm, TAKE-ALL, which, at any step, covers an element σ_i by the whole family F_i. We can easily show that the

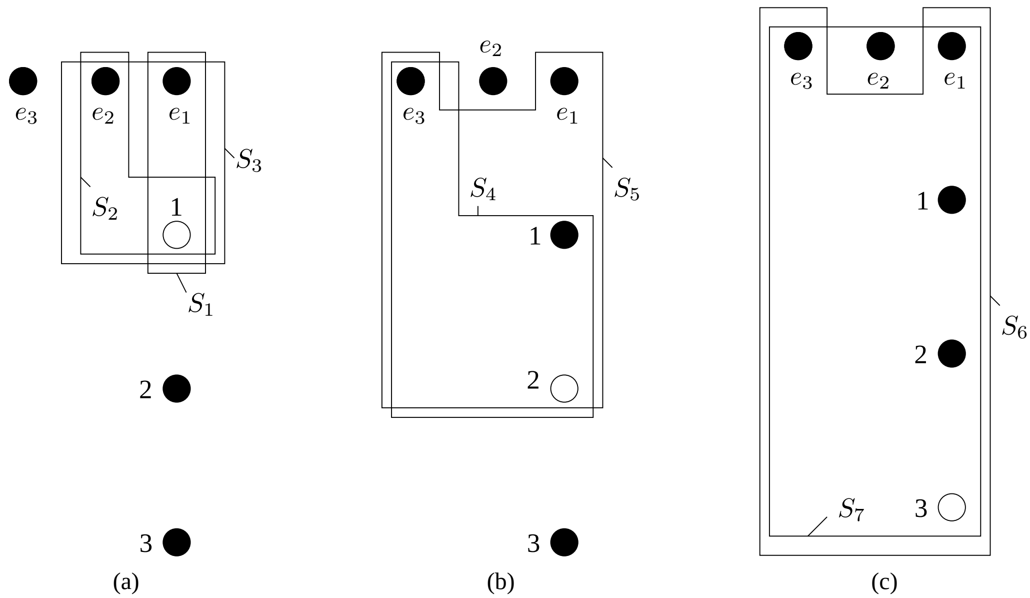

Fig. 1. An illustrative example of the proof of Proposition 1.

competitive ratio of TAKE-ALL is bounded above by f and that this bound is tight. Indeed, denote by $\sigma_1, \ldots, \sigma_k$ the critical elements of Σ, i.e., the elements having entailed inclusion of sets of $\mathcal{S}$ in $\mathcal{S}'$. Denote also by $\mathcal{S}^*$ an optimum off-line solution. Obviously, for any of the critical elements, a distinct set is needed to cover it, in any feasible cover for C; hence: $|\mathcal{S}^*| \geqslant k$. On the other hand, since, for $i = 1, \ldots, k$, $f_i \leqslant f$, $|\mathcal{S}'| \leqslant kf$. So, putting things together, a competitive ratio f is immediately derived. In order to show tightness, consider an instance with ground set $C = \{1, \ldots, n\}$ and $\mathcal{S}$, of size 2^{n-1}, consisting of all $S \subseteq \{1, \ldots, n\}$ with $1 \in S$. With an arrival sequence starting with 1, the competitive ratio of TAKE-ALL would be $2^{n-1} = f$. Finally, let us note that, from this discussion, TAKE-ALL gives a much worse competitiveness than n.

In what follows, we take into consideration a randomized algorithm and we show that also for such algorithm the achieved competitiveness is fairly poor. Let us consider Algorithm TAKE-AT-RANDOM which, when a new element σ_i is revealed, choses a set in F_i with uniform probability. In the following theorem, we show that, under model $\mathcal{M}_1$ (where the whole instance is not known in advance), the competitiveness of such algorithm cannot be much better than the competitiveness of a deterministic algorithm.

Proposition 2 *For any $\epsilon > 0$, there exists an instance of the on-line set-covering problem with n ground elements such that the expected competitive ratio of TAKE-AT-RANDOM is $\Omega(n^{1-\epsilon})$.*

Proof. For any $\epsilon > 0$, fix an integer $k > 1/\epsilon$ and let $N > 2^k$ and $N^{k+1} < 2^{N-2}/N$. Consider the instance with ground set $C = \{1, \ldots, n = N^k\}$. Family $\mathcal{S}$ contains the following three types of sets: (i) a partition of the ground set C into *class* sets $S(i) = \{j \in C : (j-1) \div N = (i-1)\}$, for $i = 1, \ldots, N^{k-1}$ (where $\div$ denotes the integer division); clearly, $|S(i)| = N$ and there exist N^{k-1} class sets; (ii) for $i \leqslant N^{k-1}$, any non-empty proper subset of $S(i)$; notice that, for any $j \in S(i)$, there exist exactly $2^{N-1}-1$ such *internal* sets that element j belongs to; (iii) the ground set C itself.

Consider the sequence such that the adversary always reveals the uncovered element of lowest index, and compute the expected value of the cover, which will be equal to the expected competitive ratio of TAKE-AT-RANDOM (equality holds, since the optimum for this instance is C).

We denote by $\mathbf{E}(q)$ the expected size of the partial solution provided by TAKE-AT-RANDOM after q elements have been revealed. Fix $m \leqslant N^{k-1}$ and $0 < p \leqslant N$. Assume that, for every $q < N(m-1)+p$:

$$\mathbf{E}(q) \geqslant \frac{q}{N} \qquad (1)$$

Let us now show that (1) remains true for $j = N(m-1)+p$. When element $j = N(m-1)+p$ has been revealed, TAKE-AT-RANDOM has to choose among the $2^{N-1}+1$ sets that j belongs to, i.e., among the ground set C, the class set $S(m)$ and the internal sets associated with j. If C has been selected, the algorithm stops. Else, $l \leqslant p$ elements get covered, then $j' = N(m-1)+p-l$ is revealed, and so on.

Since at most $N-p$ elements of $S(m)$ have already been covered, $2^{N-p}\binom{N-1}{l-1}$ different sets contain exactly l terms corresponding to elements that have not been revealed yet. Then:

$$\begin{aligned}\mathbf{E}\left(N(m-1)+p\right) &= \frac{1}{2^{N-1}+1}\left(1+2^{N-p}\sum_{l=1}^{p}\right.\\ &\qquad\left.\binom{p-1}{l-1}\left(1+\mathbf{E}\left(N(m-1)+p-l\right)\right)\right)\\ &\geqslant \frac{1}{2^{N-1}+1}\left(1+2^{N-p}\sum_{l=0}^{p-1}\binom{p-1}{l}\right.\\ &\qquad\left.\left(m+\frac{p-l-1}{N}\right)\right)\\ &\geqslant m+\frac{p}{N}-\frac{2^{N-p}}{N\left(2^{N-1}+1\right)}\left(\sum_{l=0}^{p-1}\binom{p-1}{l}(l+1)\right)\\ &\geqslant m+\frac{p}{N}-\frac{p\left(2^{N-1}\right)}{N\left(2^{N-1}+1\right)} \quad\geqslant\quad m+\frac{p}{N}-1\end{aligned}$$

The recursive relation above yields then directly $\mathbf{E}(N^k) \geqslant N^{k-1}$, i.e., $\mathbf{E}(n) = \Omega(n^{1-\epsilon})$. ■

We now give an example of construction of Proposition 2. Consider $N = 3$ and $k = 3$ (these values of N and k are not conformal with their definition but, in a first time, we use them for simplicity). Then $C = \{1,2,\ldots,27\}$ and we have: class sets: $\{1,2,3\}$, $\{4,5,6\}$, $\{7,8,9\}$, $\{10,11,12\}$, $\{13,14,15\}$, $\{16,17,18\}$, $\{19,20,21\}$, $\{22,23,24\}$ and $\{25,26,27\}$; for any class set $\{a,b,c\}$, there exist the internal sets: $\{a\}$, $\{b\}$, $\{c\}$, $\{a,b\}$, $\{a,c\}$ and $\{b,c\}$; finally, $C = \{1,2,\ldots,27\} \in \mathcal{S}$.

Let us assume that $\sigma_1 = 17$. With it will be revealed the following $2^2+1 = 5$ sets: $\{17\}$, $\{16,17\}$, $\{17,18\}$, $\{16,17,18\}$ and $\{1,2,\ldots,27\}$. The average cover for the whole instance will be of size 9, independently on the arrival sequence.

For $k = 3$, $N \geqslant 8$. Taking $N = 8$, $n = 512$. In this case, the class sets would be the partition of $\{1,2,\ldots,512\}$ into 64 subsequent 8-tuples. For any class set there would be 254 internal sets. With any element of the arrival sequence there would arrive one class set, plus 63 internal sets plus the set $\{1,2,\ldots,512\}$, i.e., $8^{3-1}+1 = 65$ sets. The average cover size would be in this case 64.

4. Model $\mathcal{M}_2$

In this section, we consider an enriched model for on-line set-covering that we call model $\mathcal{M}_2$. In this model, we assume that, together with any element σ_i in the sequence, beside the names of the sets containing σ_i, also the contents of such sets are provided. In what follows, we show that, in this new model, no on-line algorithm can achieve competitive ratio better than $\sqrt{(k^*-1)2n}/k^*$, where $k^* = |\mathcal{S}^*|$, the cardinality of an optimum off-line solution, even if the algorithm is allowed to choose more than one set at each step. Besides, we show that such performance can indeed be achieved by an algorithm which, in order to cover a newly revealed element σ_i, chooses the set in F_i covering the most of the still uncovered elements.

The basic assumption for model $\mathcal{M}_2$, that is, the knowledge of the content of the sets that cover a newly revealed item, can be justified by revisiting the second application in Section 1.. There, for any topic presented on-line, the editor should prefer the area editor that covers the broadest number of other topics not presented yet. Model $\mathcal{M}_2$ well fits such a natural requirement.

4.1. *Lower bound in model $\mathcal{M}_2$*

This section is devoted to the proof of a lower bound on the competitiveness of any algorithm in model $\mathcal{M}_2$. Central step for such a bound is the following theorem.

Theorem 1 *No on-line algorithm A for $\mathcal{M}_2$ such that, each time a not yet covered element σ_i arrives, inserts into the cover at least one set containing σ_i, can construct a solution of size $k < \sqrt{(k^*-1)2n}$ (where k^* is the size of an optimal off-line solution of the instance), even if with every arriving element, the algorithm knows the content of all sets containing it.*

Proof. Consider the following set-covering instance built, for any integers N and p, upon a ground set $C = \{x_{ij} : 1 \leqslant j \leqslant i \leqslant N\} \times \{1,\ldots,P\}$. Obviously, $|C| = n = PN(N+1)/2$. For $1 \leqslant p \leqslant P$, $|C_p| = \{x_{ij} : 1 \leqslant j \leqslant i \leqslant N\} \times \{p\}$ will be called the *p-th block* of C. A *path-set of order i from C_p*, is defined as a set containing $N-i+1$ elements $\{x_{ij_i},\ldots,x_{Nj_N}\} \times \{p\}$.

The set-system $\mathcal{S}$ of the instance contains all possible path-sets of each order i, $1 \leqslant i \leqslant N$, from C_p, for all p, $1 \leqslant p \leqslant P$. Clearly, in a block C_p there exist $N!/0!$ path-sets of order 1, $N!/1!$ path-sets of order 2, and so on and, finally, $N!/(N-1)!$ path-sets of order N, i.e., in total $N!(1+1+1/2!+\ldots+1/(N-1)!) \approx eN!$

path-sets; hence, there exist about $ePN!$ path-sets in $\mathcal{S}$. Finally, the set-system $\mathcal{S}$ is completed with an additional set Y containing all elements of C but those of one path-set of order 1 from every block C_p, $1 \leqslant p \leqslant P$, that will be specified later (hence, $|Y| = n - PN$).

As long as there exist uncovered elements, the adversary may choose to reveal an uncovered element (x_{ij}, p) from the block of the smallest index p and of the lowest possible i arriving, which will be contained only in all path-sets from C_p, of order less than or equal to i. Notice that as long as algorithm A has $r < N$ path-sets from C_p inserted to the cover, there will be at least an element $(x_{(r+1)j,p)}$ for some j, $1 \leqslant j \leqslant r+1$, not yet covered. This means also that as long as the cover computed by A contains less that PN sets, there will be at least an element (belonging to some block) still uncovered.

Suppose that after the arrival of σ_t, the size k of the cover computed by A gets greater than, or equal to, PN. Clearly, $1 \leqslant t \leqslant PN$. At time $t+1$, a new element arrives, contained in some path-sets and in Y, which can now be specified as consisting of all elements in C except from the elements of p path sets $S_1^*, \ldots, S_P^*$ of order 1, one from each block C_p, their union containing $\sigma_1, \ldots, \sigma_t$. The rest of the arrival sequence is indifferent.

Clearly, the optimum cover in this case would have been path-sets $S_1^*, \ldots, S_P^*$ together with set Y, i.e., $k^* = P+1$, while, as we have already shown, $k \geqslant PN$, with N tending to $\sqrt{2n/(k^*-1)}$ as n increases, which finally yields $k \geqslant \sqrt{(k^*-1)2n}$.

It is easy to see that the above construction can be directly generalized so that the same result holds also in the case where the on-line algorithm can insert into the cover more than one set at a time. Really, if $\sigma_{t_p} = (x_{11}, p)$ then, as long as the on-line cover contain less than N sets with elements of C_p, there exists always some $i_{\ell-1} \leqslant i_\ell \leqslant N$ and some j_{i_ℓ} for which $(x_{i_\ell j_{i_\ell}}, p)$ is still uncovered. Hence, if σ_ℓ is this element, then the algorithm will have to put some sets in the cover. Finally, the algorithm will have to put in the cover N sets for each block, i.e., PN sets in total, while the optimum will always be of size $P+1$. ■

In order to illustrate the construction of Theorem 1, consider the instance of Figure 2, with $N = 3$ and $P = 2$ (the elements of C are depicted as circles labeled by (i, j, k) for $1 \leqslant j \leqslant i \leqslant 3$, $1 \leqslant k \leqslant 2$).

The path-sets from blocks C_1, and C_2 can be thought of as paths terminating to a sink on a connected component of the directed graph of Figure 2(a).

The first elements of the arrival sequence are labeled with their order of arrival: first $(1,1,1)$ arrives, and algorithm A chooses sets $\{(1,1,1),(2,1,1),(3,1,1)\}$ and $\{(1,1,1),(2,2,1),(3,2,1)\}$ for covering it. Then the uncovered element $(3,3,1)$ arrives, so A has to cover it by, say, the set $\{(2,1,1),(3,3,1)\}$. At that moment, $N = 3$ sets with elements from C_1 have been taken into the cover by A, so the adversary reveals element $(1,1,2)$ of block C_2, i.e., an uncovered element of the block of the lowest index and of the lowest possible i (Figure 2(b)). Algorithm A covers it by including $\{(1,1,2),(2,2,2),(3,1,2)\}$ in the cover. Then, $(2,1,2)$ arrives and A uses sets $\{(2,1,2),(3,2,2)\}$, and $\{(2,1,2),(3,3,2)\}$ to cover it.

An optimal cover consists of sets $\{(1,1,1),(2,2,1),(3,3,1)\}$, $\{(1,1,2),(2,1,2),(3,1,2)\}$ and the big, shadowed set consisting of the rest of the elements, which could not have been revealed to A upon the arrival of any of the first four elements (Figure 2(c)).

By Theorem 1, the following corollary, inducing a lower bound on the competitiveness of any algorithm meeting the conditions of Theorem 1 can be immediately derived.

Corollary 1 *No on-line algorithm A for $\mathcal{M}_2$ such that, each time a not yet covered element σ_i arrives, inserts into the cover at least one set containing σ_i, can achieve competitive ratio less than $\sqrt{2n(k^*-1)}/k^*$, even if with every arriving element, the algorithm knows the contents of all sets containing it and chooses a set covering the most of the uncovered elements that are going to arrive after σ_i.*

Since the function $f(x) = (x-1)/x^2$ decreases with $x \geqslant 2$, the following corollary can be also immediately derived from Corollary 1.

Corollary 2 *No algorithm meeting the conditions of Theorem 1 can achieve competitive ratio smaller than $\sqrt{n/2}$.*

4.2. *Algorithm* `TAKE-LARGEST-ON-FUTURE-ITEMS`

Let us first consider the following algorithm (`TAKE-LARGEST`) which, at any step, covers an element σ_i with a set in F_i of maximum cardinality. Observe first that the discussion about the competitiveness of deterministic algorithms that include a specific set containing a new (uncovered) element, holds also for `TAKE-LARGEST`. Hence its competitive ratio is bounded above by n.

We show that this ratio is tight up to a constant factor. Consider the following set-covering instance:

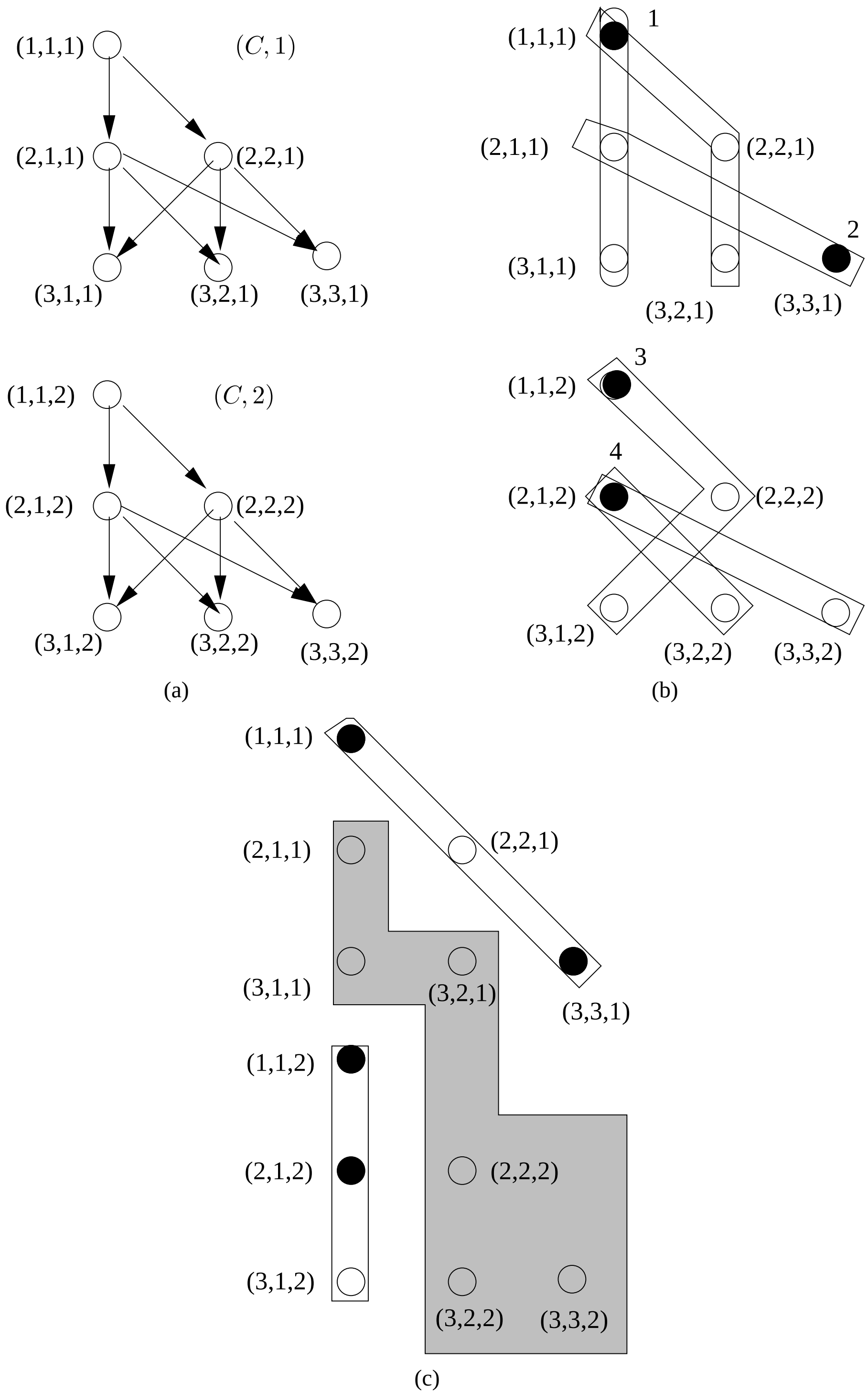

Fig. 2. An illustrative example of the proof of Theorem 1 with $N = 3$ and $P = 2$.

a ground set $C = \{1, \ldots, 2N\}$, a family of sets $\mathcal{S} = \{S_0, \ldots, S_N\}$ with $S_i = \{i, \ldots, N + i\}$. Assume an arrival sequence starting with $N, N + 1, \ldots, 2N$. Then, `TAKE-LARGEST` might include into the cover sets $S_1, \ldots, S_N$, while the optimum cover would be consisting of only S_0, S_N, thus yielding a competitive ratio of at least $N/2$.

We now assume that once an element σ_i is revealed, the contents of all sets in F_i are also revealed. In what follows, we discuss the competitive ratio achieved by algorithm `TAKE-LARGEST-ON-FUTURE-ITEMS` that exploits such information and when σ_i is revealed includes in the solution a set $\hat{S}$ that covers most of the yet uncovered items.

We show that the competitiveness of such algorithm tightly matches the lower bound of Theorem 1 and Corollary 1, and therefore can be considered an optimum algorithm in model $\mathcal{M}_2$.

As previously, we denote by k^* the size of an optimum off-line solution of an instance $(\mathcal{S}, C)$ of minimum set-covering and we set $|C| = n$.

We first give an easy upper bound for the competitive ratio of `TAKE-LARGEST-ON-FUTURE-ITEMS` that matches the lower bound of Corollary 2 up to a constant. This is the result of Proposition 3 taking into account that one can assume $k^* \geqslant 2$; otherwise ($k^* = 1$), `TAKE-LARGEST-ON-FUTURE-ITEMS` computes an optimal solution. Next we refine the analysis of the algorithm in order that its competitive ratio tightly matches the bound provided in Theorem 1.

Proposition 3 *Algorithm `TAKE-LARGEST-ON-FUTURE-ITEMS` achieves competitive ratio bounded above by* $2\sqrt{n/k^{*2}}$.

Proof. Denote by $\mathcal{S}'$ the solution computed by `TAKE-LARGEST-ON-FUTURE-ITEMS` and consider two kinds of sets chosen by the algorithm. The first kind are sets that contain at least $\sqrt{n/k^*}$ still uncovered elements; they form subfamily $\mathcal{S}'_1 \subseteq \mathcal{S}'$. The second kind are the rest of the sets chosen by the algorithm (each of them containing at most $\sqrt{n/k^*}$ uncovered elements); they form subfamily $\mathcal{S}'_2 \subseteq \mathcal{S}'$. Obviously, $\mathcal{S}' = \mathcal{S}'_1 \cup \mathcal{S}'_2$.

Observe first that the number of sets in $\mathcal{S}'_1$ is at most $\sqrt{nk^*}$, since each one covers at least $\sqrt{n/k^*}$ new elements.

We now bound from above the number of sets in $\mathcal{S}'_2$. Since all asked elements can be covered using k^* sets, then let c_i be the first element that is covered by a set in $\mathcal{S}'_2$ and is covered in the optimal solution by the set S^*_i. At the arrival of c_i, all sets containing it contain at most $\sqrt{n/k^*}$ still uncovered elements. Thus, the number of elements that belong to S^*_i and are covered by sets of $\mathcal{S}'_2$ is bounded by $\sqrt{n/k^*}$. This is true for any $1 \leqslant i \leqslant k^*$. Thus, the total number of such elements is at most $k^*\sqrt{n/k^*} = \sqrt{nk^*}$ that is also an immediate upper bound for $|\mathcal{S}'_2|$.

The discussion above derives that the total number of sets chosen by `TAKE-LARGEST-ON-FUTURE-ITEMS` is at most $2\sqrt{n/k^*}$. Dividing it by k^* immediately leads to the ratio claimed. ■

The main result of this section is Theorem 2 improving the result of Proposition 3 and establishing an upper bound in the competitiveness of `TAKE-LARGEST-ON-FUTURE-ITEMS` that tightly matches the lower bound of Theorem 1.

Theorem 2 *Algorithm `TAKE-LARGEST-ON-FUTURE-ITEMS` achieves competitive ratio bounded above by* $\sqrt{2n(k^* - 1)/k^{*2}}$.

Proof. Fix an arrival sequence $\Sigma = (\sigma_1, \ldots, \sigma_n)$, assume w.l.o.g. that $\Sigma = (1, 2, \ldots, n)$ and denote by $\mathcal{C}$ the set of its *critical elements*, i.e., the elements having entailed introduction of a set in the on-line cover $\mathcal{S}'$. In other words, critical elements of Σ are all elements c such that c was not yet covered by the cover under construction upon its arrival.

In order to make the proof's reading easier, and since a lot of new notations are used, the following example is used to illustrate proof's unraveling (see also Figure 3). Let $C = \{1, \ldots, 17\}$ be the ground set and assume that elements in C are revealed according to this ordering. Assume also that $\mathcal{C} = \{1, \ldots, 8\}$ is the sequence of critical elements and let the column sets $\{1, 9, 10, 11\}, \{2, 12, 13\}, \{3, 14, 15\}, \{4, 16\}, \{5, 17\}, \{6\}, \{7\}$ and $\{8\}$, be the cover $\mathcal{S}'$ that Algorithm `TAKE-LARGEST-ON-FUTURE-ITEMS` computes.

Fix an optimal off-line solution $\mathcal{S}^* = \{S^*_1, \ldots, S^*_{k^*}\}$ (of cardinality k^*). Any of the critical elements $c \in \mathcal{C}$ can be associated to the set of smallest index in $\mathcal{S}^*$ containing it. For any $S^*_i \in \mathcal{S}^*$, we denote by $\hat{S}^*_i$ the set of the critical elements associated with S^*_i. Obviously, for every $i \leqslant k^*$, the set of all the $\hat{S}^*_i$ is a partition of $\mathcal{C}$. For convenience, we set $h_i = |\hat{S}^*_i|$.

In our example, assume that $\mathcal{S}^* = \{\{1, 2, 8, 17\}, \{3, 4, 7, 9, 11, 12, 13\}, \{5, 6, 10, 14, 15, 16\}\}$ (i.e., the white, grey and striped entries, respectively) is the optimal off-line solution. Then, $\hat{S}^*_1 = \{1, 2, 8\}$, $\hat{S}^*_2 = \{3, 4, 7\}$ and $\hat{S}^*_3 = \{5, 6\}$.

For any $c \in \mathcal{C}$, we denote by S_c the set that has been introduced in $\mathcal{S}'$ due to the arrival of c. For any $S \subset C$ such that $c \in S$, let $\delta(c, S)$ be the set of newly covered elements that would have resulted if S was

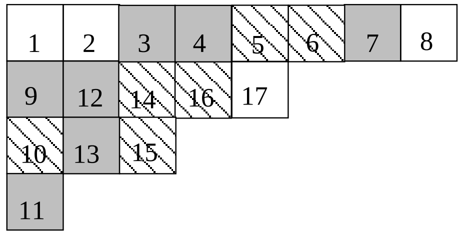

Fig. 3. An illustrative example for the proof of Theorem 2.

included in $\mathcal{S}'$ in order to cover c. In particular, $\delta(c, C)$ is the set of all elements still uncovered when c has been revealed. In our example, $\delta(3, S_3) = \{3, 14, 15\}$ while $\delta(3, S_2^*) = \{3, 4, 7\}$.

From this definition, one can get:

$$C = \bigcup_{c \in \mathcal{C}} \delta(c, S_c)$$

$$\delta(c, S) = \delta(c, C) \cap S, \quad \forall S \subset C \tag{2}$$

$$\delta(c, C) = \bigcup_{\gamma \geqslant c} \delta(\gamma, S_\gamma) \tag{3}$$

Notice that all these unions are disjoint. Since S_c has been preferred to S_i^* by the algorithm, we have, for any $c \in \mathcal{C}$ and any $i \leqslant k^*$:

$$|\delta(c, S_i^*)| \leqslant |\delta(c, S_c)| \tag{4}$$

The main point is now to find a lower bound for $|\delta(c, S_i^*)|$. Denoting by $\bar{\mathcal{C}}$ the set $C \setminus \mathcal{C}$, we get:

$$|\delta(c, S_i^*)| = |\delta(c, S_i^*) \cap \mathcal{C}| + \left|\delta(c, S_i^*) \cap \bar{\mathcal{C}}\right| \tag{5}$$

since $\delta(c, S_i^*) \subset S_i^*$, $\delta(c, S_i^*) \cap \mathcal{C}$ is a subset of $\hat{S}_i^*$. Indeed, it is the set of the elements not covered yet, i.e., the set $\{\gamma \in \hat{S}_i^* : \gamma \geqslant c\}$. Furthermore, we can combine (2), (3) and (5) to get:

$$|\delta(c, S_i^*)| \geqslant \left|\left\{\gamma \in \hat{S}_i^* : \gamma \geqslant c\right\}\right| + \sum_{\gamma > c} \left|S_i^* \cap \delta(\gamma, S_\gamma) \cap \bar{\mathcal{C}}\right| \tag{6}$$

We will now use the notation $\delta_*(c, S) = \delta(c, S) \cap \bar{\mathcal{C}}$. For instance, with respect to our example, $\delta_*(2, S_1^*) = \{17\}$.

For any S_i^*, let $c_i^1, \ldots, c_i^{h_i}$ be the elements of its critical content $\hat{S}_i^*$ ordered according to their position in the arrival sequence Σ. For instance, with respect to our guide-example, $c_1^2 = 2$ and $c_3^2 = 6$.

Clearly, for all $\ell \leqslant h_i$:

$$\left|\left\{\gamma \in \hat{S}_i^* : \gamma \geqslant c_i^\ell\right\}\right| = h_i - \ell + 1 \tag{7}$$

Let us now split $\mathcal{C}$ into two subsets, $\mathcal{C}^- = \{c \in \mathcal{C} : c < c_{k^*}^1\}$ and $\mathcal{C}^+ = \{c \in \mathcal{C} : c \geqslant c_{k^*}^1\}$. In our example, $\mathcal{C}^- = \{1, \ldots, 4\}$ and $\mathcal{C}^+ = \{5, \ldots, 8\}$.

Although it will be necessary to keep a tight inequality for the elements of $\mathcal{C}^-$, combining (6) and (7), we write roughly, for any $c \in \mathcal{C}^+$ and for any $\ell \leqslant h_i$:

$$\left|\delta\left(c_i^\ell, S_i^*\right)\right| \geqslant h_i - \ell + 1 \tag{8}$$

Summing up inequalities, for $c \in \mathcal{C}$, and taking into account that the sets $\{\delta(c, S_c)\}_{c \in \mathcal{C}}$ form a partition of C, we get:

$$\begin{aligned} n &= \sum_{\substack{i \leqslant k^* \\ \ell \leqslant h_i}} \left|\delta\left(c_i^\ell, S_{c_i^\ell}\right)\right| \\ &= \sum_{c_i^\ell \in \mathcal{C}^-} \left|\delta\left(c_i^\ell, S_{c_i^\ell}\right)\right| + \sum_{c_i^\ell \in \mathcal{C}^+} \left|\delta\left(c_i^\ell, S_{c_i^\ell}\right)\right| \\ &\geqslant \sum_{c_i^\ell \in \mathcal{C}^-} \left|\delta\left(c_i^\ell, S_i^*\right)\right| + \sum_{c_i^\ell \in \mathcal{C}^+} \left|\delta\left(c_i^\ell, S_i^*\right)\right| \end{aligned} \tag{9}$$

expression (9) holding because of (4).

According to previous assumptions, we bound above tightly the first term and roughly the second one, using respectively (6) and (8):

$$\begin{aligned} n &\geqslant \sum_{c_i^\ell \in \mathcal{C}^-} \left(h_i - \ell + 1 + \sum_{\gamma > c_i^\ell} |S_i^* \cap \delta_*(\gamma, S_\gamma)|\right) + \\ &\quad \sum_{c_i^\ell \in \mathcal{C}^+} (h_i - \ell + 1) \\ &\geqslant \sum_{c_i^\ell \in \mathcal{C}} (h_i - \ell + 1) + \sum_{c_i^\ell \in \mathcal{C}^-} \sum_{\gamma > c_i^\ell} |S_i^* \cap \delta_*(\gamma, S_\gamma)| \end{aligned}$$

Recall that, for any $c_i^\ell \in \mathcal{C}^-$, $\gamma \in \mathcal{C}^+$ implies $\gamma > c_i^\ell$. This allows us to switch the indices in the double sum. Furthermore, notice that the last term remains unchanged when ℓ varies. Hence:

$$\begin{aligned} n &\geqslant \sum_{c_i^\ell\in\mathcal{C}} (h_i-\ell+1) + \sum_{c_i^\ell\in\mathcal{C}^-}\sum_{\gamma\in\mathcal{C}^+} |S_i^*\cap\delta_*(\gamma,S_\gamma)| \\ &\geqslant \sum_{i\leqslant k^*}\sum_{\ell\leqslant h_i}(h_i-\ell+1) + \\ &\quad \sum_{\gamma\in\mathcal{C}^+}\sum_{i\leqslant k^*}\sum_{\ell:c_i^\ell\in\mathcal{C}^-}|S_i^*\cap\delta_*(\gamma,S_\gamma)| \\ &\geqslant \frac{1}{2}\sum_{i\leqslant k^*}h_i(h_i+1) + \\ &\quad \sum_{\gamma\in\mathcal{C}^+}\sum_{i\leqslant k^*}\left|\left\{c\in\hat{S}_i^*: c<c_{k^*}^1\right\}\right| |S_i^*\cap\delta_*(\gamma,S_\gamma)| \end{aligned} \tag{10}$$

In order to feel what this step of the proof means, let us focus on our guiding example. According to (8), we must cover two additional elements when 1 or 3 arrive, and one additional element when 2, 4 or 5 are revealed. Furthermore, according to (10), and since 17 belongs to $S_1^*\cap h_*(5,S_5)$, we have to cover one element again for each critical element in S_1^* lower than 5, i.e., for 1 and 2. Thus, $h(1,S_1)\geqslant 4$, $h(2,S_2)\geqslant 3$, $h(3,S_3)\geqslant 3$, $h(4,S_4)\geqslant 2$, $h(5,S_5)\geqslant 2$; so, $n\geqslant 17$.

Fix now i_0 such that $\epsilon=\min_{i<k^*}\{|\{c\in\hat{S}_i^*: c<c_{k^*}^1\}|\}$ is realized for i_0. Notice that, by definition, no element of $\hat{S}_i^*$ belongs to $\mathcal{C}^-$ if and only if $i=k^*$. In other words, $\epsilon\geqslant 1$. Recall also that the sets S_i^* form a cover of C. Thus:

$$\begin{aligned} 2n &\geqslant \sum_{i\leqslant k^*}h_i(h_i+1) + 2\sum_{\gamma\in\mathcal{C}^+}\epsilon\left|\bigcup_{i\leqslant k^*}(S_i^*\cap\delta_*(\gamma,S_\gamma))\right| \\ &\geqslant \sum_{i\leqslant k^*}h_i(h_i+1)+2\epsilon\sum_{\gamma\in\mathcal{C}^+}|\delta_*(\gamma,S_\gamma)| \end{aligned}$$

Once again, we combine (4) and (8) to get a lower bound, denoted by h_i^+, for the quantity $|\delta(\gamma,S_\gamma)| = |\{c\in\hat{S}_i^*: c\geqslant\gamma\}|$. Recall that $\delta_*(\gamma,S_\gamma)=\delta(\gamma,S_\gamma)\cap\bar{\mathcal{C}}=\delta(\gamma,S_\gamma)\setminus\{\gamma\}$. Then:

$$\sum_{\gamma\in\mathcal{C}^+}|\delta_*(\gamma,S_\gamma)| = \sum_{i\leqslant k^*}\sum_{\ell:c_i^\ell\in\mathcal{C}^+}\left|S_i^*\cap\delta_*\left(c_i^\ell,S_{c_i^\ell}\right)\right| \tag{11}$$

$$\begin{aligned} &\geqslant \sum_{i\leqslant k^*}\sum_{\ell:c_i^\ell\in\mathcal{C}^+}\left(\left(h_i^+-1\right)-\ell+1\right) \\ &\geqslant \frac{1}{2}\sum_{i\leqslant k^*}\left(h_i^+-1\right)h_i^+ \end{aligned} \tag{12}$$

From the definition of i_0 we get $h_{i_0}^+ +\epsilon = h_{i_0}$. In (12), we will keep only the terms with indices k^* and i_0:

$$\begin{aligned} 2n &\geqslant \sum_{i\leqslant k^*}h_i(h_i+1)+\epsilon\sum_{i\leqslant k^*}\left(h_i^+-1\right)h_i^+ \\ &\geqslant \sum_{i\leqslant k^*}h_i^2+\epsilon\left(h_{k^*}^2+(h_{i_0}-\epsilon)^2\right) \end{aligned} \tag{13}$$

Since the sets $\hat{S}_i^*$ form a partition of $\mathcal{C}$, we minimize the last sum in (13) under the constraint $\sum_{i\leqslant k^*}h_i=k$. This minimum is:

$$w(\epsilon)=\frac{k^2(1+\epsilon)-2\epsilon^2+\frac{\epsilon^4}{1+\epsilon}}{(1+\epsilon)(k^*-2)+2}+\frac{\epsilon^3}{1+\epsilon} \tag{14}$$

One can see from (14) that w increases with ϵ, since the numerator of its derivative is a polynomial whose coefficients are all nonnegative. Thus, $2n\geqslant w(1)\geqslant k^2/(k^*-1)$, that leads to competitiveness ratio $k/k^*\leqslant\sqrt{2n(k^*-1)/k^{*2}}$, as claimed. ■

Corollary 3 *Algorithm* `TAKE-LARGEST-ON-FUTURE-ITEMS` *achieves competitive ratio bounded above by $\sqrt{n/2}$.*

4.3. *Some remarks about* `TAKE-LARGEST-ON-FUTURE-ITEMS`

From the competitive ratio in Theorem 2, taking into account that $k^*\geqslant n/\max_{S_i\in\mathcal{S}}\{|S_i|\}$, and setting, for simplicity, $\Delta=\max_{S_i\in\mathcal{S}}\{|S_i|\}$, the following result is immediately derived.

Corollary 4 *Algorithm* `TAKE-LARGEST-ON-FUTURE-ITEMS` *achieves competitive ratio bounded above by $\sqrt{2\Delta}$.*

Also, it can be easily seen from the proof of Theorem 2 that it also works even if one assumes that the arrival sequence does not contain all the elements of C but only a part of them. So, `TAKE-LARGEST-ON-FUTURE-ITEMS` works also for the on-line model in [1] with provably competitive upper bound.

Corollary 5 *The competitive ratio of* `TAKE-LARGEST-ON-FUTURE-ITEMS` *when assumed that only a subset of C will finally be revealed is bounded above by $\sqrt{2n(k^*-1)/k^{*2}}$.*

Let us note that a ratio $O(\sqrt{n/k^*})$ is also provided in Section 5 of [1] (recall that in the model adopted in [1] the final instance is known in advance, but it is possible that only one part of it will be finally revealed) but the algorithm proposed strongly exploits the a priori knowledge of the whole instance. Besides, while the asymptotic bounds presented in [1] can be proved rather easily, the tight bounds provided in our paper require much more elaborated arguments.

We now consider the case where, whenever a yet uncovered element arrives, the algorithm is allowed to include in the cover a constant number of sets containing it and such that the number of elements yet unrevealed that belong to these sets is maximized. More precisely, consider a modification of `TAKE-LARGEST-ON-FUTURE-ITEMS` where, for a fixed number ρ, when a new ground element σ_i arrives, the ρ sets in F_i covering the most of the still uncovered elements are included in the solution. Then, the following holds.

Proposition 4 *The competitive ratio of modified `TAKE-LARGEST-ON-FUTURE-ITEMS` is bounded below by $\sqrt{\rho n}/2$.*

Proof. For some $\rho > 1$ and for some integer N, consider the following instance:

$$\mathcal{S} = \left\{X, Y, S_i^j : 1 \leqslant i \leqslant N, 1 \leqslant j \leqslant \rho\right\}$$

$$C = \bigcup_{i=1}^{N} \bigcup_{j=1}^{\rho} S_i^j \left(|C| = \rho\frac{N(N-1)}{2} + N = n\right)$$

$$X = \{x_1, \ldots, x_N\}$$

$$\left|S_i^j\right| = N - i + 1 \text{ for } i = 1, \ldots, N$$

$$S_i^j \bigcap S_l^k = \emptyset, \text{ if } i \neq l$$

$$S_i^j \bigcap S_i^k = \{x_i\}, \text{ if } j \neq k$$

$$Y = C \setminus X$$

Consider the arrival sequence where $x_1, \ldots, x_N$ are firstly revealed. `TAKE-LARGEST-ON-FUTURE-ITEMS` might include in the cover all the S_i^j's, while the optimal cover is $\{X, Y\}$. In this case, the competitive ratio is $\rho N/2$, with:

$$N = \frac{\rho - 2}{2\rho} + \sqrt{\left(\frac{\rho - 2}{2\rho}\right)^2 + 2\frac{n}{\rho}}$$

i.e., the value of the ratio is asymptotically $\sqrt{\rho n/2}$.

For example, set $\rho = 2$ and $N = 5$ and consider the instance of Figure 4. For Σ starting with x_1, x_2, x_3, x_4, x_5, the algorithm may insert to the cover the sets depicted as "rows", while the optimal cover would consist of the "column"-set $\{x_1, x_2, x_3, x_4, x_5\}$ together with the "big" set containing the rest of the elements (drawn striped in Figure 4). ■

In the weighted version of set-covering, any set S of $\mathcal{S}$ is assigned with a non-negative weight $w(S)$, and a cover $\mathcal{S}'$ of the least possible total weight $W = \sum_{S \in \mathcal{S}'} w(S)$ has to be computed. A natural modification of `TAKE-LARGEST-ON-FUTURE-ITEMS` in order to handle weighted set-covering is to put in the cover, whenever a still uncovered element arrives, a set S_i containing it that minimizes the quantity $w(S_i)/\delta(S_i)$. Unfortunately, this modification cannot perform satisfactorily. Consider, for example, an instance of weighted set-covering consisting of a ground set $C = \{x_1, \ldots, x_n\}$, and three sets, $S = C$ with $w(S) = n$, $X = \{x_1\}$ with $w(X) = 1$ and $Y = C \setminus \{x_1\}$ with $w(Y) = 0$. If x_1 arrives first, the algorithm could have chosen S to cover it, yielding a cover for the overall instance of total weight n, while the optimal cover would be $\{X, Y\}$ of total weight 1.

As a final remark, let us point out that model $\mathcal{M}_2$ and Algorithm `TAKE-LARGEST-ON-FUTURE-ITEMS` are also meaningful in the off-line setting. In such a setting the model settled here has the following meaning: an algorithm receives a permutation on the items of the ground set and must process it so that items are processed according to this order. If we restrict ourselves to algorithms that make decisions based upon the sets that cover the current item, then Algorithm `TAKE-LARGEST-ON-FUTURE-ITEMS` is quite similar, although less powerful as it has been shown, to the greedy algorithm for set-covering and achieves a non-trivial (and optimal as shown by Theorems 1 and 3) approximation ratio.

5. Discussion

In this paper we have discussed various simple on-line models for the set-covering problem. In particular we have addressed the model in which, whenever an element is revealed, no auxiliary information is provided concerning the sets that cover it beside their names (model $\mathcal{M}_1$) and the model in which the content of the sets associated with the revealed element is provided (model $\mathcal{M}_2$). In the first case, we show that no deterministic algorithm can achieve a competitive ratio better than $O(n)$ and a similar poor behavior is achieved by a randomized algorithm. In the second case an algorithm is shown, algorithm `TAKE-LARGEST-ON-FUTURE-ITEMS`, which, in order to cover a still uncovered element, chooses the set which covers the most of the still unrevealed elements. Such algorithm is the natural analogue of the greedy algorithm that is used in the off-line context and achieves the competitive ratio $O(\sqrt{(k^* - 1)2n/k^*})$

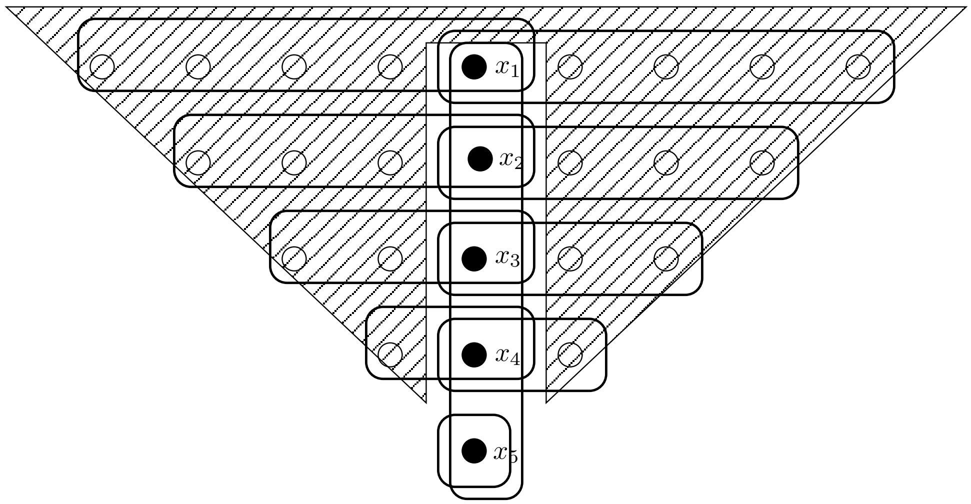

Fig. 4. A counter-example for the case where the algorithm is allowed to include a constant number of sets containing a recently arrived element.

which matches the lower bound in model $\mathcal{M}_2$. An interesting aspect of the given algorithms is their low complexity both in terms of running time and in terms of memory requirements. The memory needed by such algorithms is in fact $O(n \log m)$ and the algorithms are therefore interesting in the case of large instances. Note that this is not the case for the intensive computation implied by the model in [1].

It is important to observe that, in order to achieve the above-mentioned competitive ratio, algorithm `TAKE-LARGEST-ON-FUTURE-ITEMS` does not need to know all sets that cover the current item extensionally but it only needs to know the name of one of the sets that cover most of the yet uncovered elements. Although it is unnatural to consider that such information is provided, we stress that if this happens, say by virtue of an oracle, then our algorithm would achieve the same competitive ratio as in Theorem 2 with low complexity both in terms of running time and memory.

Let us note that the on-line models described in the paper can be extended to apply to a different but related problem, the minimum dominating set. Consistently with the model that we have adopted for the set-covering problem, our model for this latter problem is as follows. Given a graph $G(V, E)$ with $|V| = n$, assume that its vertices are revealed one-by-one. Any time a vertex σ_i is revealed, the names of its neighbors are announced.

Consider the following classical reduction from minimum dominating set to set-covering:

- $\mathcal{S} = C = V$;
- the set $S_i \in \mathcal{S}$, corresponding to the vertex $v_i \in V$, contains elements $c_{i_1}, c_{i_2}, \ldots$, of C corresponding to the neighbors $v_{i_1}, v_{i_2}, \ldots$, of v_i in G.

The set-covering instance $(\mathcal{S}, C)$ so constructed, has $|\mathcal{S}| = |C| = n$. Furthermore, it is easy to see that any set cover of size k in $(\mathcal{S}, C)$ corresponds to a dominating set of the same size in G and vice-versa. Remark also that the dominating set model just assumed on G is exactly, with respect to $(\mathcal{S}, C)$, the set-covering models handled in the paper.

Acknowledgment. The very pertinent comments and suggestions of an anonymous referee are gratefully acknowledged.

References

[1] N. Alon, B. Awerbuch, Y. Azar, N. Buchvinder, and S. Naor. The online set cover problem. In *Proc. STOC'03*, pages 100–105, 2003.

[2] V. Chvátal. A greedy-heuristic for the set covering problem. *Math. Oper. Res.*, 4:233–235, 1979.

[3] G. Gambosi, M. Protasi, and M. Talamo. Preserving approximation in the min-weighted set cover problem. *Discrete Appl. Math.*, 73:13–22, 1997.

[4] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278, 1974.

[5] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors,

Complexity of computer computations, pages 85–103. Plenum Press, New York, 1972.

[6] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13:383–390, 1975.

[7] 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.

[8] R. Raz and S. Safra. A sub-constant error probability low-degree test and a sub-constant error probability PCP characterization of NP. In *Proc. STOC'97*, pages 475–484, 1997.

[9] P. Slavík. A tight analysis of the greedy algorithm for set cover. In *Proc. STOC'96*, pages 435–441, 1996.

[10] D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985.

[11] O. A. Telelis and V. Zissimopoulos. Dynamic maintenance of approximate set covers. Manuscript, 2004.

Received 12 Feb 08; revised 15 Dec 08; accepted 22 Dec 08