Citation for published version (APA):
Fair Hitting Sequence problem: scheduling activities with varied frequency requirements

Serafino Cicerone¹, Gabriele Di Stefano², Leszek Gasieniec³, Tomasz Jurdzinski⁴, Alfredo Navarra⁵, Tomasz Radzik⁶, and Grzegorz Stachowiak⁴

¹ University of L’Aquila, Italy, sera.fino.cicerone@univaq.it
² University of L’Aquila, Italy, gabriele.distefano@univaq.it
³ University of Liverpool, U.K., l.a.gasieniec@liverpool.ac.uk
⁴ University of Wroclaw, Poland, tju@cs.uni.wroc.pl
⁵ University of Perugia, Italy, alfredo.navarra@unipg.it
⁶ King’s College London, U.K., tomasz.radzik@kcl.ac.uk

Abstract. Given a set \( V = \{v_1, \ldots, v_n\} \) of \( n \) elements and a family \( S = \{S_1, S_2, \ldots, S_m\} \) of (possibly intersecting) subsets of \( V \), we consider a scheduling problem of perpetual monitoring (attending) these subsets. In each time step one element of \( V \) is visited, and all sets in \( S \) containing \( v \) are considered to be attended during this step. That is, we assume that it is enough to visit an arbitrary element in \( S_j \) to attend to this whole set. Each set \( S_j \) has an urgency factor \( h_j \), which indicates how frequently this set should be attended relatively to other sets. Let \( t^{(j)}_i \) denote the time slot when set \( S_j \) is attended for the \( i \)-th time. The objective is to find a perpetual schedule of visiting the elements of \( V \), so that the maximum value \( h_j \left( t^{(j)}_{i+1} - t^{(j)}_i \right) \) is minimized. The value \( h_j \left( t^{(j)}_{i+1} - t^{(j)}_i \right) \) indicates how urgent it was to attend to set \( S_j \) at the time slot \( t^{(j)}_{i+1} \).

We call this problem the Fair Hitting Sequence (FHS) problem, as it is related to the minimum hitting set problem. In fact, the uniform FHS, when all urgency factors are equal, is equivalent to the minimum hitting set problem, implying that there is a constant \( c_0 > 0 \) such that it is \( \text{NP}-\text{hard to compute} \ (c_0 \log m)\)-approximation schedules for FHS.

We demonstrate that scheduling based on one hitting set can give poor approximation ratios, even if an optimal hitting set is used. To counter this, we design a deterministic algorithm which partitions the family \( S \) into sub-families and combines hitting sets of those sub-families, giving \( O(\log^2 m)\)-approximate schedules. Finally, we show an LP-based lower bound on the optimal objective value of FHS and use this bound to derive a randomized algorithm which with high probability computes \( O(\log m)\)-approximate schedules.

Keywords: scheduling; periodic maintenance; hitting set; approximation algorithms

* The work has been supported in part by the European project “Geospatial based Environment for Optimisation Systems Addressing Fire Emergencies” (GEO-SAFE), contract no. H2020-691161, by the Italian National Group for Scientific Computation GNCs-IndAM, by Networks Sciences and Technologies (NeST) initiative at University of Liverpool, and by the Polish National Science Center (NCN) grant 2017/25/B/ST6/02010.
1 Introduction

The combinatorial problem studied in this paper is a natural extension of the following perpetual scheduling proposed in [11]. Nodes $v_1, v_2, \ldots, v_n$ of a network need to be indefinitely monitored (visited) by a mobile agent according to their urgency factors $h_1, h_2, \ldots, h_n$, which indicate how often each node should be visited relatively to other nodes in the network. The (current) urgency indicator of node $v_i$ is defined as $t \cdot h_i$, where $t$ is the time which has elapsed since the last visit to this node. The objective of scheduling visits to nodes is to minimise the maximum value ever observed on the urgency indicators. Two variants of this problem were considered in [11]. In the discrete variant the time needed to visit each node is assumed to be uniform, corresponding to a single round of the monitoring process. The continuous variant assumes that the nodes are distributed in a geometric space and the time required to move to, and attend, the next node depends on the current location of the mobile agent.

In this paper we consider a generalization of the discrete variant of the perpetual scheduling, where the emphasis is on monitoring a given family of sets $S = \{S_1, S_2, \ldots, S_m\}$, which are (possibly intersecting) subsets of the set of $n$ network nodes. We assume that it is enough to visit an arbitrary node in a set $S_j$ to attend to this whole set. Moreover, by visiting a node we assume that all sets in $S$ containing this node are attended. As in the perpetual scheduling problem studied in [11], we schedule visits to nodes, but these visits are now only means of attending to sets $S_j$ and the urgency factors $h_1, h_2, \ldots, h_m$ are associated with these sets, not with the nodes.

This generalization of the perpetual scheduling problem is motivated by dissemination (or collection) of information across different, possibly overlapping, communities in social (media) networks. Participants of such a network can provide access to all communities to which they belong. While a lot of work has been done on recognition/detection of communities, starting with the seminal studies presented in [12, 18], much less is known about efficient ways of informing or monitoring such communities, especially when they are highly overlapping and dynamic and have their own frequency requirements. One way of modeling such problems is to decide whom and when to contact to ensure regular, but proportionate to the requirements, access to all communities.

Other scenarios motivating our scheduling problem arise in the context of overlapping sensor or data networks. Consider overlapping networks $S_1, S_2, \ldots, S_m$ and access nodes $v_1, v_2, \ldots, v_n$. Each node $v_i$ is an access node of one or more networks $S_{i1}, S_{i2}, \ldots, S_{ik}$, $k \geq 1$. In the context of our abstract scheduling problem, these overlapping networks correspond to overlapping communities of the previous scenario. Each network $S_j$ has a specified access rate $h_j > 0$, which indicates how often this network should be accessed relative to other networks. If an access node $v_i$ is used at the current time slot, then all networks $S_{i1}, S_{i2}, \ldots, S_{ik}$ containing $v_i$ are accessed during this time slot. Accessing a network can be thought of, for example, as gathering data from that network, or providing some other service, maintenance or update for that network. We want to find an infinite schedule $A = (v_{q1}, \ldots, v_{qt}, \ldots)$, where $v_{qt}$ is the access
node used in the time slot \( t \geq 1 \), so that each network is accessed as often as possible and in a fair way according to the specified access rates.

*Fair Hitting Sequence problem.* We formalize the objective of the regular and fair access to networks \( S_1, S_2, \ldots, S_m \) in the following way. When progressing through a schedule \( A \), if a network \( S_j \) was accessed for the last time at a time slot \( t' \), then the value \( h_j (t - t') \) indicates the urgency of accessing this network at the current time slot \( t > t' \). We refer to this value as the *urgency indicator* of network \( S_j \), or simply as the (current) urgency or the *height* of \( S_j \). The urgency indicator of \( S_j \) grows with the rate \( h_j \) over the time when \( S_j \) is not accessed and is reset to 0 when \( S_j \) is accessed. Hence we will refer to numbers \( h_j \) also as *growth rates* (of urgency indicators). We want to find a schedule which minimizes the rate \( S \) through a schedule \( A \) at the current time slot.

That is, \( \max \{ h_j (t - t') : 1 \leq j \leq m \} \) is the maximum value, or the maximum height, of the urgency indicator of \( S_j \), and is called the *height of \( S_j \). The height of such optimal (access) nodes of this network (or all members of this community). The simplest, and trivial, instance of the FHS problem is when \( |S_j \cap W| = 1 \), for each \( 1 \leq j \leq m \), and \( W \) has the minimum size among all subsets of \( V \) with this property. The height of such optimal schedule is equal to \(|W|\). NP-hardness of the *minimum hitting set problem,*
which is equivalent to the minimum cover set problem, implies NP-hardness of the more general FHS problem. The natural greedy algorithm for the minimum hitting-set problem, which selects in each iteration a node hitting (belonging to) the maximum number of the remaining sets $S_j$, gives an $O(\log m)$-approximate hitting set. On the other hand, it is known that there is a constant $c_0 > 0$ such that finding a $(c_0 \log m)$-approximate hitting set is NP-hard [21]. This implies NP-hardness of $(c_0 \log m)$-approximation for the more general FHS problem.

Continuing with the case of uniform growth rates, if all sets $S_j$ have size 2, then such an instance is represented by the graph $G = (V, E)$, where $E = \{S_1, S_2, \ldots, S_m\}$. In this case the FHS problem becomes a problem of efficient monitoring of the edges of graph $G$ (by visiting veritces of $G$), which is equivalent to the vertex cover problem.

Another non-trivial special case of the FHS problem is when $S_j = \{v_j\}$, for each $1 \leq j \leq n$, but the access rates $h_j$ are non-uniform. This is the perpetual scheduling problem considered in [4, 8, 11]. If we further assume that all input parameters $h_j$ are inverses of positive integer numbers, then the question whether there exists a schedule of height not greater than 1 is known as the Pinwheel scheduling problem [14].

We are interested in deriving good approximation algorithms for the FHS problem. While schedules are defined as infinite sequences, it can be shown that there is always an optimal schedule which has a periodic form $B_{init}(B_{\text{period}})^*$, where $B_{init}$ and $B_{\text{period}}$ are finite schedules (see e.g. [1]). The period of a periodic optimal schedule can have exponential length, but our approximate algorithms compute in polynomial time schedules with periods polynomial in $m$.

\textbf{Our results.} If we denote by $A(W)$ the schedule obtained by repeating the same hitting set $W$, then the height of $A(W)$ is at most $h_{\text{max}}|W|$, where $h_{\text{max}} = \max_{1 \leq j \leq m}\{h_j\}$. Actually it is possible to show instances for which the $A(W)$ schedule is only $\Theta(m/\log m)$ approximate. To get better schedules, we have to handle the variations in the growth rates $h_j$. In Section 2, we present simple $O(\log^2 m)$-approximate schedules. Such schedules are obtained by partitioning the whole family of sets $S_j$ into $O(\log m)$ sub-families of sets which have similar growth rates, and by combining $O(\log m)$-approximate hitting sets of these sub-families. To improve further the approximation ratio of computed schedules, we first derive in Section 3 a lower bound on the height of any schedule. This lower bound can be viewed as the optimal solution to a fractional version of the FHS problem. Then we show in Section 4 a randomized algorithm which uses the optimal fractional solution to compute schedules which are $O(\log m)$-approximate with probability at least $1 - 1/m$.

\textbf{Previous related results.} Several constant approximation algorithms for the discrete variant and $O(\log n)$ approximation for the continuous variant of this perpetual scheduling problem are discussed in [11] and further work on this problem is presented in [4, 8]. In [4], the authors consider monitoring by two agents of $n$ nodes located on a line and requiring different frequencies of visits. The authors
provide several approximation algorithms concluding with the best currently
known $\sqrt{3}$-approximation.

The perpetual scheduling problem considered in [4, 8, 11] is closely related
to periodic scheduling [22], general Pinwheel scheduling [2, 3], periodic Pinwheel
scheduling [14, 15], and to other problems motivated by Pinwheel scheduling
[20]. This problem is also related to several classical algorithmic problems which
focus on monitoring and mobility. These include the Art Gallery Problem [19]
and its dynamic alternative called the k-Watchmen Problem [17, 23]. In further
work on fence patrolling [5, 6] the authors focus on monitoring vital (possibly
disconnected) parts of a linear environment where each point is expected to
be visited with the same frequency. The authors of [7] study monitoring linear
environments by agents prone to faults.

2 Deterministic $O(\log^2 m)$-approximate schedules

In this section, we show a deterministic approximation algorithm for the FHS
problem. The algorithm exploits the properties of schedules which are based on
hitting sets.

2.1 Algorithm based on hitting sets

We first formalize an observation that if there is not much variation among
the growth rates of the sets, then the minimum hitting set gives a good ap-
proximate solution. Consider an input instance with $h_{\text{max}} \leq C \cdot h_{\text{min}}$, where
$h_{\text{min}} = \min_{1 \leq j \leq m} \{h_j\}$ and $C \geq 1$ is a parameter. Let $W_{\text{opt}}$ be a minimum
hitting set and compare the heights of the schedule $A(W_{\text{opt}})$ and an optimal
schedule $A_{\text{opt}}$. We note that an optimal schedule exists since the schedule $A(V)$
(the round-robin schedule $(v_1, v_2, \ldots, v_n)$) has height $nh_{\text{max}}$ and all (infinitely
many) schedules with heights at most $nh_{\text{max}}$ have heights in the finite set
${ih_j : j = 1, 2, \ldots, m, i \cdot \text{ positive integer, } ih_j \leq h_{\text{max}} \cdot n}$.

Let $[1, t]$ be the shortest initial time interval in schedule $A_{\text{opt}}$ when each set
is accessed at least once. We have $t \geq |W_{\text{opt}}|$, since the set of nodes used in the
first $t$ time slots in schedule $A_{\text{opt}}$ is a hitting set. Let $S_j$ be any set accessed for
the first time in schedule $A_{\text{opt}}$ at time $t$. We have

$$\text{Height}(A(W_{\text{opt}})) \leq h_{\text{max}}|W_{\text{opt}}| \leq C \cdot h_j \cdot |W_{\text{opt}}| \leq C \cdot h_j \cdot t \leq C \cdot \text{Height}(A_{\text{opt}}),$$

where the last inequality follows from the fact that in schedule $A_{\text{opt}}$, the height
of set $S_j$ (that is, the height of its urgency indicator) at time $t$ is equal to
$h_j t$. Thus $A(W_{\text{opt}})$ is a $C$-approximate schedule. If $W_{\text{apx}}$ is a $D$-approximate
hitting set ($|W_{\text{apx}}| \leq D \cdot |W_{\text{opt}}|$), then a similar argument shows that $A(W_{\text{apx}})$
is a $(CD)$-approximate schedule. This and the $O(\log m)$ approximation of the
greedy algorithm for the hitting set problem give the following lemma.

Lemma 1. If $W_{\text{apx}}$ is a $D$-approximate hitting set, then the schedule $A(W_{\text{apx}})$
is $(DH_{\text{max}}/h_{\text{min}})$-approximate. There is a polynomial-time algorithm which com-
putes $O((\log m)h_{\text{max}}/h_{\text{min}})$-approximate schedules for the FHS problem.
For the growth rates, we take similar growth rates. More precisely, we partition the whole family of sets \( S = \{S_1, S_2, \ldots, S_m\} \) into the following \( k_{\text{max}} = \lceil \log m \rceil + 1 \) families.

\[
F_k = \{S_j : h_{\text{max}}/2^k < h_j \leq h_{\text{max}}/2^{k-1}\}, \quad \text{for } k = 1, 2, \ldots, k_{\text{max}} - 1,
\]

\[
F_{k_{\text{max}}} = \{S_j : h_j \leq h_{\text{max}}/2^{k_{\text{max}}-1}\}.
\]

Let \( W_k \) be a \( D \)-approximate hitting set for the family \( F_k \), \( 1 \leq k \leq k_{\text{max}} - 1 \), and let \( W_{k_{\text{max}}} \) be any hitting set for the family \( F_{k_{\text{max}}} \) such that \( |W_{k_{\text{max}}}| \leq |F_{k_{\text{max}}}| \).

For \( 1 \leq k \leq k_{\text{max}} - 1 \), the schedule \( A(W_k) \), which repeats the same permutation of \( W_k \), is a \((2D)\)-approximate schedule for the family \( F_k \) (from Lemma 1). The schedule \( A(W_{k_{\text{max}}} \), which repeats the same permutation of \( W_{k_{\text{max}}} \), is a schedule for the family \( F_{k_{\text{max}}} \) with height at most \( |W_{k_{\text{max}}}| (h_{\text{max}}/2^{k_{\text{max}}-1}) \leq m (h_{\text{max}}/2^{k_{\text{max}}-1}) \leq 2h_{\text{max}} \leq 2 \cdot H(A_{\text{opt}}) \). Therefore the schedule \( A \) which interleaves the \( k_{\text{max}} \) schedules \( A(W_1), A(W_2), \ldots, A(W_{k_{\text{max}}-1}), A(W_{k_{\text{max}}} \) \) is a \((2Dk_{\text{max}})\)-approximate schedule for the whole family \( S \). This is because for each \( 1 \leq k \leq k_{\text{max}} \), the lengths of the periods in the schedule \( A(W_k) \) between the consecutive accesses to the same set \( S_j \in F_k \) increase \( k_{\text{max}} \) times in the schedule \( A \). (Set \( S_j \) may have some some additional accesses in \( A \), which come from other schedules \( A(W_{k'}), k' \neq k \)).

**Theorem 1.** The schedule \( A \) constructed above using \( D \)-approximate hitting sets is \( O(D \log m) \) approximate.

**Corollary 1.** There is a polynomial-time algorithm which computes \( O(\log^2 m) \)-approximate schedules for the FHS problem.

### 2.2 A tight example for using \( \log m \) hitting sets

We showed in Section 2.1 that the schedule \( A \) which is based on \( \log m \) hitting sets computed separately for the groups of sets with similar growth rates is \( O(D \log m) \)-approximate, where \( D \) is an upper bound on the approximation ratio of the used hitting sets (Theorem 1). We provide now an instance of FHS such that even if optimal hitting sets are used, the schedule \( A \) is only \( \Theta(\log m) \)-approximate.

Consider the following instance for the FHS problem, illustrated in Figure 1. Given an integer \( t > 0 \), let \( m = 2^t - 1 \) be the number of sets. The sets are defined as follows:

- \( S_{t,i} = \{v_{t,i}\} \), for each \( i = 1, 2, \ldots, \frac{m+1}{2} = 2^{t-1} \);
- \( S_{t,1} = S_{t+1,2} \cup S_{t+1,4} \cup \{v_{t,1}\} \), for each \( t = t-1, t-2, \ldots, 1 \) and for each \( i = 1, 2, \ldots, \frac{m+1}{2^{t+1}} = 2^{t-1} \).

For the growth rates, we take \( h(S_{t,i}) = \frac{1}{2^t} \) for each \( t = t-1, \ldots, 1 \) and \( i = 1, 2, \ldots, \frac{m+1}{2^{t+r+t}} = 2^{t-1} \).
Fig. 1. An instance of the FHS problem for algorithm $A$ defined in Theorem 1.

On this instance, we now compare the performance of an optimum schedule with the schedule $A$ defined in Theorem 1.

Any schedule must cover separately the sets $S_{t,i}$, $i = 1, \ldots, \frac{m+1}{2}$, since each of these sets is a singleton containing a different element $v_{t,i}$. Thus, the height of any schedule is at least $\frac{1}{2}$. A schedule given by an interleaved round-robin on elements $v_{t,i}$, $i = 1, \ldots, \frac{m+1}{2}$ achieves this lower bound. In fact, these elements form a hitting set for the whole family $S$. By interleaved schedule, we mean that the elements are picked according to a permutation which ensures that each set $S_{t,i}$ at level $t$ is served every $2^{t-1}$ time slots, so it grows to the maximum height of $\frac{1}{2}$. For instance, for $t = 4$, the schedule can be $v_{4,1}, v_{4,5}, v_{4,3}, v_{4,7}, v_{4,2}, v_{4,6}, v_{4,4}, v_{4,8}$.

On the other hand, the schedule $A$ constructed as in Section 2.1 has height $\log(m+1)$. Indeed, the $\frac{m+1}{2}$ sets at level $t$ have growth rates $\frac{1}{2^t}$, and each of these sets is served every $\frac{m+1}{2} t$ time slots, giving the height

$$\frac{m+1}{2} \cdot t \cdot \frac{1}{2^t} = \frac{t}{2} = \log(m+1).$$

The heights of the sets at other levels are never greater than $\frac{1}{2}$, so the approximation ratio of schedule $A$ is $\Theta(\log m)$.

3 A lower bound via the fractional solution

We derive a lower bound on the height of any schedule $A$ of the FHS problem.

Consider a schedule $A = (v_{q_1}, \ldots, v_{q_t}, \ldots)$ in which each $S_j$, $1 \leq j \leq m$, is accessed infinitely many times (otherwise the schedule has infinite height) and take a large time slot $T$. We look at the first $T$ slots of schedule $A$, that is, at the schedule $A[T] = (v_{q_1}, v_{q_2}, \ldots, v_{q_T})$. For $i = 1, 2, \ldots, n$, let $z_i$ denote the fraction of the time slots $1, 2, \ldots, T$ when the node $v_i$ is used, that is, $z_i = |\{1 \leq t \leq T | v_i \text{ is used in slot } t\}|$.
by increasing $T$ up to infinity (so $I(T)$ increases to infinity) we conclude that

$$\text{Height}(A) \geq X_{\text{opt}}.$$ (10)
The linear program \((P)\) can be viewed as giving the optimal solution for the following fractional variant of the FHS problem. For the discrete FHS problem, a schedule \(A\) can be represented by binary values \(y_{i,t} \in \{0, 1\}\), \(1 \leq i \leq n, t \geq 1\), with \(y_{i,t} = 1\) indicating that node \(v_i\) is used in the time slot \(t\). For the fractional variant of FHS, a schedule is represented by numbers \(0 \leq y_{i,t} \leq 1\) indicating the fraction of commitment during the time slot \(t\) to node \(v_i\). (Think about the nodes being dealt with during the time period \((t - 1, t]\) concurrently, with the fraction \(y_{i,t}\) of the total effort spent on node \(v_i\).) In both discrete and fractional cases we require that \(\sum_{t=1}^{m} y_{i,t} = 1\), for each time slot \(t \geq 1\). For the discrete variant, the time slot \(t_i^{(j)}\) when \(S_j\) is accessed for the \(i\)-th time is the time slot \(\tau\) such that

\[
\sum_{t=1}^{\tau} (y_{j_1,t} + y_{j_2,t} + \cdots + y_{j_{q(i)},t}) = i.
\]

For the fractional variant, the time \(t_i^{(j)}\) when the \(i\)-th “cycle” of access to \(S_j\) is completed (and the urgency indicator of \(S_j\) is reset to 0) is the fractional time \(\tau + \delta\), where \(\tau\) is a positive integer and \(0 \leq \delta < 1\), such that

\[
\sum_{t=1}^{\tau} (y_{j_1,t} + y_{j_2,t} + \cdots + y_{j_{q(i)},t}) + \delta (y_{j_1,\tau+1} + y_{j_2,\tau+1} + \cdots + y_{j_{q(i)},\tau+1}) = i.
\]

In both cases, the fraction of the period \((0, T]\) when a node \(v_i\) is used is equal to \(z_i = (\sum_{t=1}^{T} y_{i,t}) / T\) and (3)–(7) and (10) apply. For the fractional variant, the schedule \(y_{i,t} = x_i^{*}\), for \(1 \leq i \leq n\) and \(t \geq 1\), where \((x_1^{*}, x_2^{*}, \ldots, x_n^{*}, X_{\text{opt}})\) is an optimal solution of \((P)\), has the optimal (minimum) height \(X_{\text{opt}}\).

## 4 Randomized \(O(\log m)\)-approximate algorithm

We use an optimal solution \((x_1^{*}, x_2^{*}, \ldots, x_n^{*}, X_{\text{opt}})\) of linear program \((P)\) to randomly select nodes for the first \(T = \Theta(m)\) slots of a schedule \(A\), so that with high probability each set \(S_j\) is accessed at least once during each period \([t + 1, t + \tau_j] \subseteq [1, T]\), where \(\tau_j = \Theta((X_{\text{opt}} / h_j) \log n)\). Thus during the first \(T\) slots of the schedule, the heights of the urgency indicators remain \(O(X_{\text{opt}} \log n)\). The full (infinite) schedule keeps repeating the schedule from the first \(T\) slots. In our calculations we assume that \(m \geq m_0\), for a sufficiently large constant \(m_0\).

We take \(T = 2m\) and construct a random schedule \(A_R = (v_{q_1}, v_{q_2}, \ldots, v_{q_{2m}})\) for \(T\) time slots in the following way. We put aside the even time slots for some deterministic assignment of nodes. Specifically, for each time slot \(t \geq 2j, j = 1, 2, \ldots, m\), we (deterministically) take for the node \(v_{q_t}\) for this time slot an arbitrary node in \(S_j\). This way we guarantee that each set \(S_j\) is accessed at least once when the schedule \(A_R\) is followed. For each odd time slot \(t\), \(1 \leq t < T\), node \(v_{q_t}\) is a random node selected according to the distribution \((x_1^{*}, x_2^{*}, \ldots, x_n^{*})\) and independently of the selection of other nodes. Thus for each odd time slot \(t \in [1, T]\) and for each node \(v_i \in V\), \(\Pr(v_{q_t} = v_i) = x_i^{*}\).
Lemma 2. The random schedule $A_R$ has the properties that each set $S_j$, $j = 1, 2, \ldots, m$, is accessed at least once and with probability at least 1 − 1/m, $\text{Height}(A_R) \leq (5 \ln m)X_{\text{opt}}$.

Proof. The first property is obvious from the construction. We show that with probability at least 1 − 1/m, no urgency indicator grows above $(5 \ln m)X_{\text{opt}}$. A set $S_j$ with the rate growth $h_j < (2.5X_{\text{opt}} \ln m)/m$ cannot grow above the height $h_jT < 5X_{\text{opt}} \ln m$, so it suffices to look at the growth of the sets $S_j$ with $h_j \geq (2.5 \cdot X_{\text{opt}} \ln m)/m$. Observe that $X_{\text{opt}} \geq h_{\text{max}} = \max\{h_1, h_2, \ldots, h_m\}$, from (8).

Let $J \subseteq \{1, 2, \ldots, m\}$ be the set of indices of the sets $S_j$ for which $h_j \geq (2.5 \cdot X_{\text{opt}} \ln m)/m$. For each $j \in J$ and for each odd time slot $t \in [1, T]$, the probability that set $S_j$ is accessed during this time slot is equal to $x_{j_1}^* + x_{j_2}^* + \cdots + x_{j_q}^* \geq h_j/X_{\text{opt}}$. In each period $[t, t + \tau - 1] \subseteq [1, T]$ of $\tau$ consecutive time slots, there are at least $\lceil \tau/2 \rceil$ odd time slots, so the probability that $S_j$ is not accessed during this period is at most $(1 - h_j/X_{\text{opt}})^{\lceil \tau/2 \rceil}$. We take $\tau_j = 5(X_{\text{opt}}/h_j) \ln m$ (observe that $\ln m \leq \tau_j \leq T$) and use the union bound over all $j \in J$ and all $[t, t + \tau_j - 1] \subseteq [1, T]$ to conclude that the probability that there is a set $S_j$, $j \in J$, which is not accessed during consecutive $\tau_j$ time slots (and its urgency indicator goes above $(5 \ln m)X_{\text{opt}}$) is at most

$$T \cdot \sum_{j \in J} \left(1 - \frac{h_j}{X_{\text{opt}}}\right)^{(\tau_j-1)/2} \leq T \cdot \sum_{j \in J} \left(1 - \frac{h_j}{X_{\text{opt}}}\right)^{2.4(X_{\text{opt}}/h_j) \ln m} \leq 2m \cdot e^{-2.4 \ln m} \leq \frac{1}{m}. \quad \Box$$

Theorem 2. For the infinite schedule $A_R^*$ which keeps repeating the same random schedule $A_R$ (all copies are the same), $\text{Height}(A_R^*) \leq (10 \ln m)X_{\text{opt}}$ with probability at least 1 − 1/m.

Proof. With probability at least 1 − 1/m, $\text{Height}(A_R) \leq (5 \ln m)X_{\text{opt}}$ (Lemma 2). Assuming that $\text{Height}(A_R) \leq (5 \ln m)X_{\text{opt}}$, we show that $\text{Height}(A_R^*) \leq (10 \ln m)X_{\text{opt}}$.

Let $T = 2m$ be the length of the schedule $A_R$. We consider an arbitrary set $S_j$ and show that its height is never greater than $(10 \ln m)X_{\text{opt}}$ when the schedule $A_R^*$ is followed. Since $S_j$ is accessed in $A_R$ at least once, the height of $S_j$ under the schedule $A_R^*$ is the same at the end of the time slots $kT$, for all positive integers $k$ (and is equal to $h_j \left(T - t_{\text{last}}^{(j)}\right)$, where $t_{\text{last}}^{(j)}$ is the last time slot in $A_R$ when $S_j$ is accessed). The maximum height of $S_j$ during the period $[1, T]$ is at most $(5 \log m)X_{\text{opt}}$. For each integer $k \geq 1$, the maximum height of set $S_j$ during the period $[kT + 1, (k + 1)T]$ is at most the height of $S_j$ at the end of time slot $kT$, which is at most $(5 \ln m)X_{\text{opt}}$, plus the maximum growth of $S_j$ under the schedule $A_R$, which is again at most $(5 \ln m)X_{\text{opt}}$. Thus the height of $S_j$ is never greater than $(10 \ln m)X_{\text{opt}}$. \quad \Box
5 Concluding remarks

We studied the Fair Hitting Sequence problem, showing its wide range of applications. We provide both deterministic and randomized approximation algorithms, with approximation ratios of $O(\log^2 m)$ and $O(\log m)$, respectively. These upper bounds should be compared with the lower bound of $\Omega(\log m)$ on the approximation ratio of polynomial-time algorithms, which is inherited from the well-known minimum hitting set problem. As a natural question one may ask whether it is possible to provide a deterministic algorithm with approximation ratio guarantee of $O(\log m)$. Due to the deep relation shown for FHS with the hitting set problem, one may be interested in understanding whether introducing some restriction on the sets might result in better approximation ratios. For instance, interesting cases might be when the size of each set $S_j$ is bounded, when each element is contained in a bounded number of sets, or when the intersection of each pair of sets is bounded. In particular, when the size of each set is two, then the sets can be seen as edges of a graph, as mentioned in Section 1, and one may consider special graph topologies.

When we consider more than two elements per set, then instead of graphs we actually deal with hypergraphs. In the finite hypergraph setting, a (minimal) hitting set of the edges is called a (minimal) transversal of the hypergraph [9]. Fixed-parameter tractability results have been obtained for the related transversal hypergraph recognition problem with a wide variety of parameters, including vertex degree parameters, hyperedge size or number parameters, and hyperedge intersection or union size parameters [13]. Concerning special classes of hypergraph, it is known that the transversal recognition is solvable in polynomial time for special cases of acyclic hypergraphs [9, 10]. These results for transversal of hypergraphs may be useful in further study of the FHS problem.

Furthermore, some variants of the FHS problem may be interesting from the theoretical or practical point of view. For instance, one may consider the elements embedded in the plane and the time required by a visiting agent to move from one element to another defined by the distance between those elements. In such setting, it may be useful to consider the following geometric version of the hitting set problem given in [16]. Given a set of geometric objects and a set of points, the goal is to compute the smallest subset of points that hit all geometric objects. The authors of [16] provide $(1 + \epsilon)$-approximation schemes for the minimum geometric hitting set problem for a wide class of geometric range spaces. It would be interesting to investigate how these results could be applied in the wider context of the FHS problem. Finally, further investigations can come from the variant where sets dynamically evolve, as it would be expected in the context of evolving communities in a social network.

References