Fair Hitting Sequence problem: scheduling activities with varied frequency requirements*

Serafino Cicerone[1], Gabriele Di Stefano[2], Leszek Gasieniec[3], Tomasz Jurdzinski[4], Alfredo Navarra[5], Tomasz Radzik[6], and Grzegorz Stachowiak[7]

[1] University of L'Aquila, Italy, serafino.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, tomasz.jurdzinski@uwr.edu.pl
[5] University of Perugia, Italy, alfredo.navarra@unipg.it
[6] King's College London, U.K., tomasz.radzik@kcl.ac.uk
[7] University of Wroclaw, Poland, gst@ii.uni.wroc.pl

Abstract. Given a set $V = \{v_1, \ldots, v_n\}$ of n elements and a family $\{S_1, S_2, \ldots, S_m\}$ 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 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 $t_i^{(j)}$ denote the time slot when set S_j is attended for the i-th time. The objective is to find a perpetual schedule of of visiting the elements of V, *i.e.* an infinite sequence of elements to visit in consecutive steps, so that the maximum value $h_j(t_{i+1}^{(j)} - t_i^{(j)})$ is minimized. The value $h_j(t_{i+1}^{(j)} - t_i^{(j)})$ indicates how urgent it was to attend to set S_j at the time slot $t_{i+1}^{(j)}$. 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 (all urgency factors are equal) is equivalent to the minimum hitting set problem, implying that there exists a constant $c_0 > 0$ such that it is NP-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 of sets S_j into sub-families and combines hitting sets of those sub-families, giving $O(\log^2 m)$-approximate schedules. Finally, we show a lower bound on the optimal objective value of FHS and use this bound to derive a randomized algorithm which computes $O(\log m)$-approximate schedules with probability $1 - 1/m$.

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 and by the Italian National Group for Scientific Computation GNCS-INdAM.

1 Introduction

The combinatorial problem studied in this paper is a natural extension of perpetual scheduling proposed in [11], where n network nodes need to be indefinitely monitored (visited) by a mobile agent, according to known frequencies. More precisely, in this problem all n nodes $v_1, v_2, \ldots, v_n$ have *urgency factors* $h_1, h_2, \ldots, h_n$, respectively, which indicate how often each node should be visited relatively to other nodes in the network. Two variants of the problem of scheduling visits to nodes were considered in [11]. In the first *discrete variant* the time needed to visit each node is assumed to be uniform and it corresponds to a single round of the monitoring process. The second *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 both cases, when t is the time which has elapsed since the last visit to node v_i, the *urgency indicator* of this node shows the value $t \cdot h_i$. The objective of scheduling visits to nodes is to minimise the maximum value ever observed on the urgency indicators.

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.

In this paper we consider a generalization of the perpetual scheduling problem, where the emphasis is on perpetual monitoring of a given family of sets $\{S_1, S_2, \ldots, S_m\}$, 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 to attend the whole set. Moreover, by visiting a node we assume that *all* sets containing this node are attended. Similarly to the discrete variant of the perpetual scheduling problem studied in [11], here we also schedule visits to nodes, but now the visits to nodes are just means to attend the sets S_j and the urgency factors $h_1, h_2, \ldots, h_m$ are associated with these sets, not with the nodes.

This scheduling problem is motivated by dissemination (or collection) of information across different, possibly overlapping, communities in social (media) networks. A participant of the overall network can provide access to all communities to which this participant belongs. 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 the communities are highly overlapping and dynamic, and have their own frequency requirements (urgency factors). 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 (that is, sharing some nodes) 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 $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$, $k \geq 1$. In the context of our abstract scheduling problem, overlapping networks and overlapping communities are analogous entities. Each network S_j has a specified required *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 $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ 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 some other service, maintenance or update performed on that network, depending on the application. We want to find a schedule $\mathcal{A} = (v_{q_1}, \ldots, v_{q_t}, \ldots)$, where v_{q_t} 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.

We formalize the objective of the regular and fair access to networks in the following way. When progressing through a schedule $\mathcal{A}$, if a network S_j was accessed for the last time at a time slot t', then the number $h_j\,(t - t')$ indicates the urgency of accessing this network at the current time slot $t > t'$. For brevity, we refer to this number 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 maximum $h_j\left(t_{i+1}^{(j)} - t_i^{(j)}\right)$, over all networks S_j, $j = 1, 2, \ldots, m$ and all $i \geq 0$, where $t_i^{(j)}$ is the i-th time slot when network S_j is accessed (setting $t_0^{(j)} \equiv 0$). That is, $h_j\left(t_{i+1}^{(j)} - t_i^{(j)}\right)$ is the height of S_j at the time when this network is (about to be) accessed for the $(i+1)$-st time.

For a given schedule $\mathcal{A} = (v_{q_1}, v_{q_2}, \ldots)$ and $1 \leq j \leq m$, the number

$$\mathit{Height}(\mathcal{A}, j) = \sup\left\{h_j\left(t_{i+1}^{(j)} - t_i^{(j)}\right) :\ i \geq 0\right\} \tag{1}$$

is the maximum value, or the maximum height, of the urgency indicator of S_j, when schedule $\mathcal{A}$ is followed, and the number

$$\mathit{Height}(\mathcal{A}) = \max\left\{\mathit{Height}(\mathcal{A}, j) : 1 \leq j \leq m\right\} \tag{2}$$

is the maximum height of any urgency indicator and is called the *height of schedule* $\mathcal{A}$. We want to find an *optimal schedule* $\mathcal{A}_{\mathrm{opt}}$ which minimizes (2). We refer to this problem as the *Fair Hitting Sequence* (FHS) problem, and show below that it includes the *hitting set* problem as a special case. We say that a schedule $\mathcal{A}$ is ρ-approximate, if $\mathit{Height}(\mathcal{A}) \leq \rho \cdot \mathit{Height}(\mathcal{A}_{\mathrm{opt}})$.

We denote by $V = \{v_1, v_2, \ldots, v_n\}$ the set of all access nodes (or all participants in a social network), which from now on will be simply referred to as nodes, and we identify each network (or a community) S_j with the set $\{v_{j_1}, v_{j_2}, \ldots, v_{j_q}\} \subseteq V$ of all (access) nodes of this network (or all members of this community). The simplest, and trivial, instance of the problem FHS is when $m = n$, $S_j = \{v_j\}$ and $h_j = 1$, for all $1 \leq j \leq n$. For this instance a schedule is optimal if, and only if, it is a repetition of the same permutation of V. The height of such a schedule is equal to n.

A still special, but more interesting and non-trivial, case is when sets S_j are arbitrary, with possibly $m \neq n$, but all h_j remain equal to 1. It is not difficult to see that for such instances of FHS a schedule is optimal if, and only if, it is a repetition of the same permutation of the same minimum-size *hitting set* $W \subseteq V$. That is, $|S_j \cap W| \geq 1$, 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 $\mathcal{B}_{init}(\mathcal{B}_{period})^*$, where $\mathcal{B}_{init}$ and $\mathcal{B}_{period}$ 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.

If we denote by $\mathcal{A}(W)$ the schedule obtained by repeating the same hitting set W, then the height of $\mathcal{A}(W)$ is at most $h_{\max}|W|$, where $h_{\max} = \max_{1 \leq j \leq m}\{h_j\}$. We show in Appendix instances for which the $\mathcal{A}(W)$ 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 high probability.

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 approximate solution. Consider an input instance with $h_{\max} \le Ch_{\min}$, where $h_{\min} = \min_{1 \le j \le m}\{h_j\}$ and $C \ge 1$ is a parameter. Let $W_{\rm opt}$ be a minimum hitting set and compare the heights of the schedule $\mathcal{A}(W_{\rm opt})$ and an optimal schedule $\mathcal{A}_{\rm opt}$. We note that an optimal schedule exists since the schedule $\mathcal{A}(V)$ (the round-robin schedule $(v_1, v_2, \ldots, v_n)^*$) has height $nh_{\max}$ and all (infinitely many) schedules with heights at most $nh_{\max}$ have heights in the finite set $\{ih_j : j = 1, 2, \ldots, m,\ i$ - positive integer, $ih_j \le h_{\max} \cdot n\}$.

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

$$\mathit{Height}(\mathcal{A}(W_{\rm opt})) \le h_{\max}|W_{\rm opt}| \le C\, h_j\, |W_{\rm opt}| \le C\, h_j\, t \le C \cdot \mathit{Height}(\mathcal{A}_{\rm opt}),$$

where the last inequality follows from the fact that in schedule $\mathcal{A}_{\rm opt}$, the height of set S_j (that is, the height of its urgency indicator) at time t is equal to $h_j t$. Thus $\mathcal{A}(W_{\rm opt})$ is a C-approximate schedule. If $W_{\rm apx}$ is a D-approximate hitting set ($|W_{\rm apx}| \le D \cdot |W_{\rm opt}|$), then a similar argument shows that $\mathcal{A}(W_{\rm apx})$ 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_{apx} is a D-approximate hitting set, then the schedule $\mathcal{A}(W_{apx})$ is $(Dh_{\max}/h_{\min})$-approximate. There is a polynomial-time algorithm which computes $O((\log m)h_{\max}/h_{\min})$-approximate schedules for the FHS problem.*

If there is considerable variation in the growth rates h_j, then the schedule $\mathcal{A}(W_{\rm opt})$, which relies on one common minimum hitting set, can be far from optimal (see the example in Appendix). To get a better approximation, we consider separately sets with similar growth rates. More precisely, we partition the whole family of sets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ into the following $k_{\max} = \lfloor \log m \rfloor + 1$ families.

$$\begin{aligned}\mathcal{F}_k \quad &= \{S_j : h_{\max}/2^k < h_j \le h_{\max}/2^{k-1}\}, \quad \text{for } k = 1, 2, \ldots, k_{\max} - 1,\\ \mathcal{F}_{k_{\max}} &= \{S_j : h_j \le h_{\max}/2^{k_{\max}-1}\}.\end{aligned}$$

Let W_k be a D-approximate hitting set for the family $\mathcal{F}_k$, $1 \le k \le k_{\max} - 1$, and let $W_{k_{\max}}$ be any hitting set for the family $\mathcal{F}_{k_{\max}}$. For $1 \le k \le k_{\max} - 1$, the schedule $\mathcal{A}(W_k)$, which repeats the same permutation of W_k, is a $(2D)$-approximate schedule for the family $\mathcal{F}_k$. Therefore the schedule $\mathcal{A}'$ which is the interleaving of the schedules $\mathcal{A}(W_k)$, for $1 \le k \le k_{\max} - 1$, is a $(2D \log m)$-approximate schedule for the family of sets $\bigcup_{k=1}^{k_{\max}-1} \mathcal{F}_k$. This is because the lengths of the gaps in the schedule $\mathcal{A}(W_k)$ between the consecutive accesses to a set $S_j \in \mathcal{F}_k$ increase $k_{\max} - 1$ times in the schedule $\mathcal{A}'$ (some additional accesses to S_j in $\mathcal{A}'$ may be coming from other schedules $\mathcal{A}(W_{k'})$, $k' \neq k$). The schedule $\mathcal{A}(W_{k_{\max}})$, which repeats the same permutation of $W_{k_{\max}}$, is a schedule for the family $\mathcal{F}_{k_{\max}}$ with height at most $m\left(h_{\max}/2^{k_{\max}-1}\right) \le 2h_{\max} \le 2 \cdot H(\mathcal{A}_{\rm opt})$. Therefore the schedule $\mathcal{A}$ which interleaves schedules $\mathcal{A}(W_1), \mathcal{A}(W_2), \ldots, \mathcal{A}(W_{k_{\max}-1}), \mathcal{A}(W_{k_{\max}})$ is a $2D(\log m{+}1)$-approximate schedule for the whole family of sets S_j.

Theorem 1. *The schedule $\mathcal{A}$ 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 $\mathcal{A}$ 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 $\mathcal{A}$ is only $\Theta(\log m)$-approximate.

Consider the following instance for the FHS problem, as shown in Figure 1. Given a large enough 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 $i = 1, 2, \ldots, \frac{m+1}{2}$;
- $S_{\ell,i} = S_{\ell+1,2i-1} \cup S_{\ell+1,2i} \cup \{v_{\ell,i}\}$, for each $\ell = t-1, t-2, \ldots, 1$ and for each $i = 1, 2, \ldots, \frac{m+1}{2^{t-\ell+1}}$.

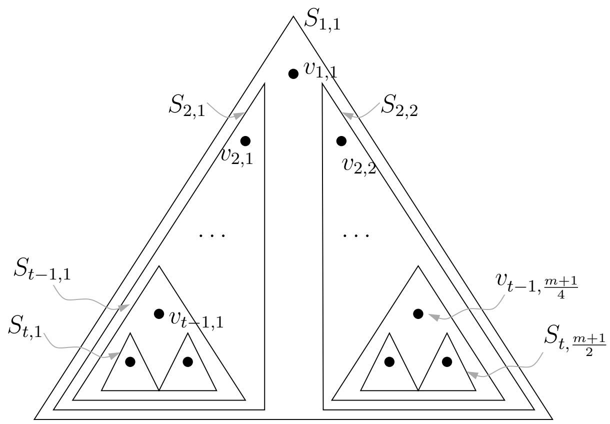

Fig. 1. An instance of the FHS problem for algorithm $\mathcal{A}$ defined in Theorem 1.

For the growth rates, we take $h(S_{\ell,i}) = \frac{1}{2^\ell}$ for each $\ell = t, t-1, \ldots, 1$ and $i = 1, 2, \ldots, \frac{m+1}{2^{t-\ell+1}}$.

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

The optimum is given by an interleaved round-robin schedule on elements $v_{t,i}$, $i = 1, ..., \frac{m+1}{2}$. In fact, such vertices represent by construction a hitting set for the provided instance, and any solution must cover the singletons $S_{t,i}$, $i = 1, ..., \frac{m+1}{2}$, each containing a different $v_{t,i}$. By interleaved schedule, we mean the vertices are not picked in sequence but in such a way, given a generic level ℓ, each set $S_{\ell,i}$ is served every $\frac{m+1}{2^{t-\ell+1}}$ times. For instance, by considering $t = 4$, the corresponding schedule will be $v_{4,1}$, $v_{4,5}$, $v_{4,3}$, $v_{4,7}$, $v_{4,2}$, $v_{4,6}$, $v_{4,4}$, $v_{4,8}$. This ensures to keep the maximum height of each level at the same value $\frac{1}{2}$. Hence such a solution provides a maximum height of $\frac{1}{2}$.

By applying algorithm $\mathcal{A}$, instead, one obtains a maximum height of $\frac{\log m}{2}$. In fact, there are $\frac{m+1}{2}$ sets at level t, with growing rate of $\frac{1}{2^t}$, each one served every t times, which gives:

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

The approximation ratio is then $O(\log m)$ which is tight.

3 A lower bound via the fractional solution

We derive a lower bound on the height of any schedule $\mathcal{A}$ of the FHS problem. Consider a schedule $\mathcal{A} = (v_{q_1}, \ldots, v_{q_t}, \ldots)$ 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 $\mathcal{A}$, that is, at the schedule $\mathcal{A}[T] = (v_{q_1}, v_{q_2}, \ldots, v_{q_T})$. 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, $z_i = |\{1 \le t \le T : v_{q_t} = v_i\}|/T$. For $j = 1, 2, \ldots, m$, let $1 \le t_1^{(j)} < t_2^{(j)} < \cdots < t_{I(j,T)}^{(j)} \le T$ be the time slots in the period $[1, T]$ when S_j is accessed. We assume that T is large enough so that for each $1 \le j \le m$, $I(j, T) \ge 1$, that is, each S_j is accessed at least once in the period $[1, T-1]$. Defining $t_0^{(j)} = 0$ and $t_{I(j,T)+1}^{(j)} = T$, the maximum height of $S_j = \{v_{j_1}, v_{j_2}, \ldots, v_{j_{q(j)}}\}$ in the period $[1, T]$ is

$$\begin{aligned} \mathit{Height}(\mathcal{A}[T], j) &= \max\left\{h_j\left(t_i^{(j)} - t_{i-1}^{(j)}\right) : 1 \le i \le I(j,T)+1\right\} && (3)\\ &\ge \frac{h_j}{I(j,T)+1} \sum_{i=1}^{i=I(j,T)+1} \left(t_i^{(j)} - t_{i-1}^{(j)}\right) = \frac{h_j T}{I(j,T)+1} && (4)\\ &= \frac{h_j}{z_{j_1} + z_{j_2} + \cdots + z_{j_{q(j)}}} \, \frac{I(j,T)}{I(j,T)+1}. && (5) \end{aligned}$$

Inequality (4) simply says that the maximum of $I(j, T) + 1$ numbers is at least their mean value. The equality on the last line above holds because z_{j_r} is the fraction of the time slots $1, 2, \ldots, T$ when node v_{j_r} is used, so $z_{j_1} + z_{j_2} + \cdots + z_{j_q}$ is the fraction of the time slots $1, 2, \ldots, T$ when S_j is accessed, which is equal to $I(j, T)/T > 0$. For the height of schedule $\mathcal{A}$, we have

$$\begin{aligned} &\mathit{Height}(\mathcal{A}) \ \ge && (6)\\ &\ge \mathit{Height}(\mathcal{A}[T]) \ \equiv\ \max\{\mathit{Height}(\mathcal{A}[T], j) :\ j = 1, 2, \ldots, m\}\\ &\ge \left(1 - \frac{1}{I(T)+1}\right) \max\left\{\frac{h_j}{z_{j_1} + z_{j_2} + \cdots + z_{j_{q(j)}}} \ :\ j = 1, 2, \ldots, m\right\}, && (7) \end{aligned}$$

where $I(T) = \min_{1 \le j \le m}\{I(j, T)\}$ is the minimum number of times any S_j is accessed in the period $[1, T]$.

Consider the following linear program. (To get an equivalent proper linear program, substitite X with $1/Z$ and maximize Z.)

$$\begin{aligned} (\mathcal{P}) \quad & \text{minimize } \ X;\\ & \text{subject to:}\\ & \quad x_1 + x_2 + \cdots + x_n = 1,\\ & \quad x_{j_1} + x_{j_2} + \cdots + x_{j_{q(j)}} \ge h_j/X, \quad \text{for each } j = 1, 2, \ldots, m, && (8)\\ & \quad x_i \ge 0, \quad \text{for } i = 1, 2, \ldots, n,\\ & \quad X > 0. && (9) \end{aligned}$$

Comparing Inequalities (7) with Inequalities (8), we see that by setting x_1, x_2, $\ldots, x_n$ to numbers $z_1, z_2, \ldots, z_n$ and X to $\mathit{Height}(\mathcal{A})/\left(1 - \frac{1}{I(T)+1}\right)$, we satisfy all constraints of this linear program. Thus denoting by X_{opt} the minimum feasible value of X in this linear program, we have $\mathit{Height}(\mathcal{A}) \ge X_{\mathrm{opt}}\left(1 - \frac{1}{I(T)+1}\right)$, and by increasing T to infinity (so $I(T)$ increases to infinity) we conclude that

$$\mathit{Height}(\mathcal{A}) \ge X_{\mathrm{opt}}. \tag{10}$$

The linear program $(\mathcal{P})$ can be viewed as giving the optimal solution for the following fractional variant of the FHS problem. For the discrete FHS problem, a schedule $\mathcal{A}$ can be represented by binary values $y_{i,t} \in \{0,1\}$, $1 \le i \le n$, $t \ge 1$, 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 $0 \le y_{i,t} \le 1$ 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 $\sum_{i=1}^{n} y_{i,t} = 1$, for each time slot $t \ge 1$. For the discrete variant, the time slot $t_i^{(j)}$ when S_j is accessed for the i-th time is the time slot τ such that

$$\sum_{t=1}^{\tau} \left(y_{j_1,t} + y_{j_2,t} + \cdots + y_{j_{q(j)},t} \right) = i.$$

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 τ is a positive integer and $0 \le \delta < 1$, such that

$$\sum_{t=1}^{\tau} \left(y_{j_1,t} + y_{j_2,t} + \cdots + y_{j_{q(j)},t} \right) + \delta \left(y_{j_1,\tau+1} + y_{j_2,\tau+1} + \cdots + y_{j_{q(j)},\tau+1} \right) = i.$$

In both cases, the fraction of the period $(0, T]$ when a node v_i is used is equal to $z_i = \left(\sum_{t=1}^{T} y_{i,t} \right) / T$ and (3)–(7) and (10) apply. For the fractional variant, the schedule $y_{i,t} = x_i^*$, for $1 \le i \le n$ and $t \ge 1$, where $(x_1^*, x_2^*, \ldots, x_n^*, X_{\text{opt}})$ is an optimal solution of $(\mathcal{P})$, has the optimal (minimum) height X_{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 $(\mathcal{P})$ to randomly select nodes for the first $T = \Theta(m)$ slots of a schedule $\mathcal{A}$, 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 $m \ge m_0$, for a sufficiently large constant m_0.

We take $T = 2m$ and construct a random schedule $\mathcal{A}_R = (v_{q_1}, v_{q_2}, \ldots, v_{q_T})$ 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 $t = 2j$, $j = 1, 2, \ldots, m$, 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 $\mathcal{A}_R$ is followed. For each odd time slot t, $1 \le t \le T$, 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 $v_i \in V$, $\mathbf{Pr}(v_{q_t} = v_i) = x_i^*$.

Lemma 2. *The random schedule $\mathcal{A}_R$ has the properties that each set S_j, $j = 1, 2, \ldots, m$, is accessed at least once and with probability at least $1 - 1/m$, Height$(\mathcal{A}_R) \leq (5 \ln m) X_{opt}$.*

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

Let $J \subseteq \{1, 2, \ldots, m\}$ be the set of indices of the sets S_j for which $h_j \geq (2.5 \cdot X_{\mathrm{opt}} \ln m)/m$. 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 $x^*_{j_1} + x^*_{j_2} + \cdots + x^*_{j_q} \geq h_j / X_{\mathrm{opt}}$. In each period $[t, t + \tau - 1] \subseteq [1, T]$ of τ consecutive time slots, there are at least $\lfloor \tau/2 \rfloor$ odd time slots, so the probability that S_j is not accessed during this period is at most $(1 - h_j/X_{\mathrm{opt}})^{\lfloor \tau/2 \rfloor}$. We take $\tau_j = 5(X_{\mathrm{opt}}/h_j) \ln m$ (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 τ_j time slots (and its urgency indicator goes above $(5 \ln m) X_{\mathrm{opt}}$) is at most

$$
\begin{aligned}
T \cdot \sum_{j \in J} \left(1 - \frac{h_j}{X_{\mathrm{opt}}}\right)^{(\tau_j - 1)/2} &\leq T \cdot \sum_{j \in J} \left(1 - \frac{h_j}{X_{\mathrm{opt}}}\right)^{2.4 (X_{\mathrm{opt}}/h_j) \ln m} \\
&\leq 2m \cdot e^{-2.4 \ln m} \leq \frac{1}{m}. \qquad \square
\end{aligned}
$$

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

Proof. With probability at least $1 - 1/m$, *Height*$(\mathcal{A}_R) \leq (5 \ln m) X_{\mathrm{opt}}$ (Lemma 2). Assuming that *Height*$(\mathcal{A}_R) \leq (5 \ln m) X_{\mathrm{opt}}$, we show that *Height*$(\mathcal{A}^*_R) \leq (10 \ln m) X_{\mathrm{opt}}$.

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

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

1. S. Anily, C.A. Glass, and R. Hassin, The Scheduling of Maintenance Service, Discrete Applied Mathematics 82(1-3): 27-42 (1998).

2. M.Y. Chan and F.Y.L. Chin, General schedulers for the pinwheel problem based on double-integer reduction, IEEE Transactions on Computers, 41(6):755–768, 1992.
3. M.Y. Chan and F. Chin, Schedulers for larger classes of pinwheel instances, Algorithmica, 9(5):425–462, 1993.
4. H. Chuangpishit, J. Czyzowicz, L. Gasieniec, K. Georgiou, T. Jurdzinski, and E. Kranakis, Patrolling a Path Connecting a Set of Points with Unbalanced Frequencies of Visits, SOFSEM 2018: 367–380.
5. A. Collins, J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, D. Krizanc, R. Martin, and O. Morales Ponce, Optimal patrolling of fragmented boundaries. SPAA 2013:241–250.
6. J. Czyzowicz, L. Gasieniec, A. Kosowski, and E. Kranakis, Boundary Patrolling by Mobile Agents with Distinct Maximal Speeds, ESA 2011:701–712.
7. J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, D. Krizanc, and N. Taleb, When Patrolmen Become Corrupted: Monitoring a Graph using Faulty Mobile Robots, ISAAC 2015:343–354.
8. M. D'Emidio, G. Di Stefano, and A. Navarra, Priority Scheduling in the Bamboo Garden Trimming Problem, SOFSEM 2019, to appear.
9. T. Eiter, G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, SIAM Journal on Computing 24 (6) (1995) 1278–1304.
10. T. Eiter, G. Gottlob, K. Makino, New results on monotone dualization and generating hypergraph transversals, SIAM Journal on Computing 32 (2) (2003) 514–537.
11. L. Gasieniec, R. Klasing, Ch. Levcopoulos, A. Lingas, J. Min, and T. Radzik, Bamboo Garden Trimming Problem (Perpetual Maintenance of Machines with Different Attendance Urgency Factors), SOFSEM 2017:229–240.
12. M. Girvan and M.E.J. Newman, Community structure in social and biological networks Proc. of the National Academy of Sciences, 2002, 99 (12):7821–7826.
13. M. Hagen, Algorithmic and computational complexity issues of MONET, Dr. rer. nat., Friedrich-Schiller- Universit at Jena (2008).
14. R. Holte, L. Rosier, I. Tulchinsky, and D. Varvel, Pinwheel scheduling with two distinct numbers, Theoretical Computer Science, 100(1):105–135, 1992.
15. S.-S. Lin and K.-J. Lin, A Pinwheel Scheduler for Three Distinct Numbers with a Tight Schedulability Bound, Algorithmica, 19(4): 411–426, 1997.
16. N. H. Mustafa, S. Ray, Improved Results on Geometric Hitting Set Problems. Discrete & Computational Geometry 44(4): 883–895, 2010.
17. B. Nilsson, Guarding art galleries - methods for mobile guards. PhD thesis, Department of Computer Science, Lund University, Sweden, 1995.
18. M.E.J. Newman and M. Girvan, Finding and evaluating community structure in networks, Physical review E 69 (2), 026113, 2004.
19. S. Ntafos, On gallery watchmen in grids, Information Processing Letters, 23(2):99–102, 1986.
20. T.H. Romer and L.E. Rosier, An algorithm reminiscent of euclidean-gcd for computing a function related to pinwheel scheduling, Algorithmica, 17(1):1–10, 1997.
21. R. Raz and M. Safra, A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In Proc. of STOC, pages 475–484, 1997.
22. P. Serafini and W. Ukovich, A Mathematical Model for Periodic Scheduling Problems, SIAM Journal on Discrete Mathematics, 2(4):550–581, 1989.
23. J. Urrutia, Art gallery and illumination problems, Handbook of computational geometry, 1(1):973–1027, 2000.

Appendix: inefficiency of using one minimum hitting set

As discussed in Section 2.1, the schedule $\mathcal{A}(W_{\text{opt}})$, which is based on a minimum hitting set W_{opt}, is a $(h_{\max}/h_{\min})$-approximate schedule (Lemma 1). How bad can approximation ratios actually be, if we follow this approach? Here we show an instance with $h_{\max}/h_{\min} = \Theta(m)$, where the approximation ratio of a solution for the FHS problem based on the minimum hitting set is at least $\frac{m}{2 \log m}$.

Consider an instance I of FHS consisting of $m = 2\ell$ sets: $C_i = \{c, v''_{i-1}, v'_i\}$, $P_i = \{v'_i, v''_i\}$, $i = 1, 2, \ldots, \ell$, with $v''_0 \equiv v''_\ell$ (see Figure 2), and such that $\ell = 2^k$ and k divides ℓ.[8] By denoting $h(C_i)$ and $h(P_i)$ the growth rates of C_i and P_i, respectively, we assume $h(C_i) = 1$ and $h(P_i) = 1/\ell$, $i = 1, 2, \ldots, \ell$.

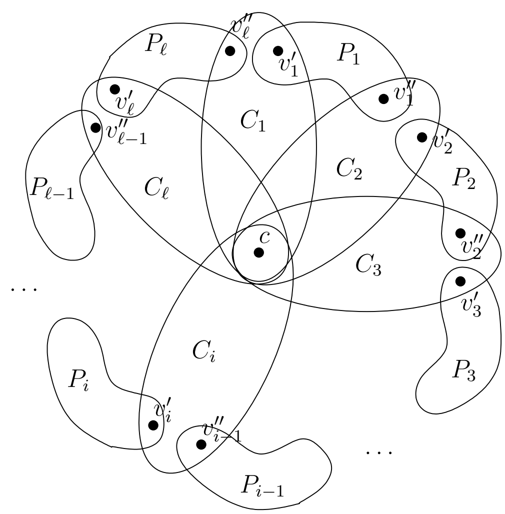

Fig. 2. An instance of the FHS problem.

The minimum hitting set HS for the instance I is given either by $\{v'_1, v'_2, \ldots, v'_\ell\}$ or $\{v''_1, v''_2, \ldots, v''_\ell\}$, which are equivalent by symmetry. All access nodes in HS being equivalent, for all schedules $\mathcal{A}(HS)$ (defined by the permutations of HS), $\mathit{Height}(\mathcal{A}(HS)) = \ell$.

Consider now the algorithm $\mathcal{A}^+$ that repeats the following scheduling:

$$(c, v'_1, v'_2, \ldots, v'_k, c, v'_{k+1}, v'_{k+1}, \ldots, v'_{2k}, \ldots\ldots, c, v'_{\ell-k+1}, v'_{\ell-k+2}, \ldots, v'_\ell).$$

Basically, scheduling $\mathcal{A}^+$ involves the access nodes of the hitting set $\{v'_1, v'_2, \ldots, v'_\ell\}$ plus the access point c not included in any hitting set. Then, it alternates c with k different elements of the hitting set.

[8] E.g., ℓ could be 2^{2^p} and $k = 2^p$, for any integer p.

Let us denote by $Height(\mathcal{A}^+, C_i)$ and $Height(\mathcal{A}^+, P_i)$ the maximum heights reached by sets C_i and P_i, respectively. Then:

$$Height(\mathcal{A}^+, C_i) = (k+1) \cdot h(C_i) = k+1,$$

as c is accessed every $k+1$ rounds,

$$Height(\mathcal{A}^+, P_i) = \left(\frac{\ell}{k} + \ell - 1\right) \cdot h(P_i) = \left(\frac{\ell}{k} + \ell - 1\right) \cdot \frac{1}{\ell} < 2,$$

as the number of slots between two visits to a generic P_i is given by the $\ell - 1$ services to sets P_j, $j \neq i$, plus $\frac{l}{k}$ accesses to c. This gives

$$Height(\mathcal{A}^+) = k+1.$$

We can now calculate the ratio between $Height(\mathcal{A}(HS))$ and $Height(\mathcal{A}^+)$:

$$\frac{Height(\mathcal{A}(HS))}{Height(\mathcal{A}^+)} = \frac{\ell}{k+1} = \frac{\ell}{\log \ell + 1} = \frac{m/2}{\log(m/2)+1} = \frac{m}{2\log m}.$$

As a consequence, it seems one should choose the access points of the schedule taking into account their 'popularity', that is how many sets the same access point serves. In the given instance I, in fact, the access point c was completely ignored by HS.

By similar arguments,one can show an instance of FHS with sets composed of just two elements each (so represented by a graph) where the solution based on the minimum hitting set (i.e. the minimum vertex cover in this case) is not helpful. One such instance of consists of $m = 3\ell$ sets: $C_i = \{c, v_i'\}$, $P_i' = \{v_i', v_i''\}$, $P_i'' = \{v_i'', v_{i+1}'\}$, $i = 1, 2, \ldots, \ell$, with $v_{l+1}' \equiv v_1'$, where $\ell = 3^k$ and k divides ℓ. Finally, $h(C_i) = 1$, $h(P_i') = 1/\ell$ and $h(P_i'') = 1/\ell$ $i = 1, 2, \ldots, \ell$.