Combinatorial Perpetual Scheduling: Existence and Computation of Low-Height Schedules

Mirabel Mendoza-Cadena
Instituto de Ciencias de la Ingeniería, Universidad de O'Higgins, Rancagua, Chile

Arturo Merino
Department of Computer Science, University of Chile, Santiago, Chile

Mads Anker Nielsen
Department of Mathematics and Computer Science, University of Cologne, Germany

Kevin Schewior
Department of Mathematics and Computer Science, University of Cologne, Germany
Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark

Abstract

This paper considers a framework for combinatorial variants of perpetual-scheduling problems. Given an independence system $(E, \mathcal{I})$, a schedule consists of an independent set $I_t \in \mathcal{I}$ for every time step $t \in \mathbb{N}$, with the objective of fulfilling frequency requirements on the occurrence of elements in E. We focus specifically on *combinatorial bamboo garden trimming*, where elements accumulate height at growth rates $g(e)$ for $e \in E$ and are reset to zero when scheduled, with the goal of minimizing the maximum height attained by any element. We assume that g is normalized so that it is a convex combination of the incidence vectors of $\mathcal{I}$.

Using the integrality of the matroid-intersection polytope, we prove that, when $(E, \mathcal{I})$ is a matroid, it is possible to guarantee a maximum height of at most 2, which is optimal. We complement this existential result with efficient algorithms for specific matroid classes, achieving a maximum height of 2 for uniform and partition matroids, and 4 for graphic and laminar matroids. In contrast, we show that for general independence systems, the optimal guaranteed height is $\Theta(\log |E|)$ and can be achieved by an efficient algorithm. For *combinatorial pinwheel scheduling*, where each element $e \in E$ needs to occur in the schedule at least every $a_e \in \mathbb{N}$ time steps, our results imply bounds on the density sufficient for schedulability.

2012 ACM Subject Classification Theory of computation → Scheduling algorithms

Keywords and phrases Perpetual Scheduling, Matroids, Bamboo Garden Trimming, Pinwheel

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.142

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2602.11826`

Funding *Mirabel Mendoza-Cadena*: Centro de Modelamiento Matemático (CMM) BASAL fund FB210005 for center of excellence from ANID-Chile.
Arturo Merino: Supported by ANID FONDECYT Iniciación No. 11251528.
Kevin Schewior: Supported in part by the Independent Research Fund Denmark, Natural Sciences, grant DFF-4283-00079B.

Acknowledgements We thank Tobias Mömke, Victor Verdugo, and José Verschae for organizing the *Santiago Summer Workshop on Combinatorial Optimization 2025*, where this research began. We especially thank Sofía Errázuriz for valuable discussions.

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 142; pp. 142:1–142:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Certain tasks such as maintenance, monitoring, or cleaning do not merely require to be accomplished once. Instead, they often require to be executed again and again under a certain frequency requirement, virtually forever. Such tasks are captured by perpetual-scheduling problems with frequency requirements, a class of problems which has recently seen a surge of interest. These problems, often appealingly clean, include the pinwheel scheduling (PS) problem (and variants) [29–31, 40] and the bamboo-garden trimming (BGT) problem (and variants) [8, 20, 36]. Kawamura's survey [32] gives a broad overview of the state of the art in this area.

To a large extent, the literature has only considered settings in which, at any time, only a single task can be scheduled. In many applications, however, it may be possible to schedule multiple tasks at the same time, subject to a combinatorial constraint. For instance, there may be multiple workers, each of which has the expertise to do a certain subset of the tasks. In this case, a set of tasks can be executed simultaneously if and only if it is an independent set of a transversal matroid. In this paper, we study a general framework for combinatorial variants of perpetual scheduling, specifically, PS and BGT. This framework essentially (cf. discussion later) unifies the few combinatorial models considered in the literature [2, 17, 24]. The framework was introduced in [11] as the *fair hitting sequence problem*. However, their work studies the problem from an approximation-algorithms perspective, which is not the main focus of the present paper.

The majority of this paper is written in terms of what we call *combinatorial BGT* (CBGT). Here, there is a ground set E of n *bamboos* and a fixed independence system $\mathcal{I} \subseteq 2^E$. Initially, each bamboo $e \in E$ has height 0 and a fixed *growth rate* $g(e) \in \mathbb{Q}_{\geq 0}$. The following two steps alternate ad infinitum: First, all bamboos grow according to their growth rates. Then, an independent set $I \in \mathcal{I}$ can be selected to be cut, i.e., all heights of bamboos in I are reduced to 0. This process leads to an infinite valid *schedule* $\pi = I_1, I_2, \ldots$, with $I_i \in \mathcal{I}$. The goal is to minimize the maximum height $h(\pi)$ that *ever* occurs among any of the bamboos. When $\mathcal{I}$ is the 1-uniform matroid, this is precisely vanilla BGT.

Note that scaling g by some factor c also just scales the height $h(\pi)$ of any schedule π by c. For this reason, it is standard for BGT to assume that g is *normalized* so that $\sum_{e \in E} g(e) = 1$. For CBGT, we generalize this by assuming that g is a convex combination of the incidence vectors of $\mathcal{I}$. That is, there exists coefficients $\lambda(I) \geq 0$ for each $I \in \mathcal{I}$ such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g = \sum_{I \in \mathcal{I}} \lambda(I) \mathbb{1}_I$, where $\mathbb{1}_I \in \{0, 1\}^E$ is the vector such that $\mathbb{1}_I(e) = 1$ iff $e \in I$ for all $e \in E$. This is essentially (again cf. discussion later) the type of normalization used in the combinatorial models in the literature [17, 24].

The main result of this paper is that, when $(E, \mathcal{I})$ is a matroid, a maximum height of at most 2 can be guaranteed. The bound is tight (as can be easily seen from an instance with $n = 2$ and growth rates $1 - \varepsilon, \varepsilon$ for vanilla BGT [16]) and generalizes the same bound for vanilla BGT [16]. While our proof only yields a pseudo-polynomial-time algorithm that produces any next cut, we give polynomial-time algorithms for uniform matroids, graphic matroids, and laminar matroids, in the latter two cases only guaranteeing a maximum height of at most 4. When $k = 1$, our algorithm for k-uniform matroids implies a simplification of the state of the art (that has appeared before in a different context [4]). We also show that, for general set systems $\mathcal{I}$, the smallest height that can be guaranteed is $\Theta(\log |E|)$. We also give a deterministic polynomial-time algorithm implementing a height-$O(\log |E|)$ schedule, which implies an $O(\log |E|)$-approximation algorithm, solving an open problem from [11].

In combinatorial PS (CPS), again an independence system $(E, \mathcal{I})$ is given, and each element e of the ground set E has a *period* $a(e) \in \mathbb{N}$. An instance is called *schedulable* if there exists an infinite valid schedule $I_1, I_2, \dots$ with $I_i \in \mathcal{I}$ such that all periods are satisfied, i.e., for each $e \in E$ and $t \in \mathbb{N}$, it holds that $e \in \bigcup_{i=t}^{i+a(e)-1} I_i$. Analogously to the above normalization, we define the *density* of an instance as the minimum ρ such that there exist $\lambda(I) \geq 0$ for all $I \in \mathcal{I}$ with $\sum_{i\in\mathcal{I}} \lambda(I) = \rho$ and $\sum_{I\in\mathcal{I}:e\in I} \lambda(I) = 1/a(e)$ for all $e \in E$. Our results for CBGT imply *density bounds*, i.e., bounds on the density that guarantees schedulability, the quantity that has been the main object of study for vanilla PS [32].

1.1 Our Contribution

In this exposition, we focus on CBGT. Our main result is the following.

▶ **Theorem 3.1.** *For any CBGT instance $(E, \mathcal{I}, g)$ where $(E, \mathcal{I})$ is a matroid, there exists an infinite valid schedule π for $(E, \mathcal{I}, g)$ with $h(\pi) < 2$.*

A first approach could be the following: Reduce the problem to vanilla BGT by identifying the independent sets $I \in \mathcal{I}$ with bamboos in the new instance and interpreting the coefficients $\lambda(I)$ for all $I \in \mathcal{I}$ as growth rates. Then an optimal valid schedule π' for the vanilla-BGT instance can be interpreted as a valid schedule π for the CBGT instance. Unfortunately, this leads to an arbitrarily large height, even when $(E, \mathcal{I})$ is a uniform matroid: Let $n = |E|$ be even, let $\mathcal{I} = \{I \subseteq E \mid |I| \leq n/2\}$, and set $\lambda(I) = 1/\binom{n}{n/2}$ for all $I \in \mathcal{I}$ with $|I| = n/2$ (and $\lambda(I) = 0$ for all other I), which implies $g(e) = 1/2$ for all $e \in E$. Note that, on this instance, π starts with an arbitrary permutation of $\mathcal{I}$. In particular, for some fixed $e^\star$, π may first schedule all $\binom{n}{n/2}/2$ sets that do not contain $e^\star$, leading to a height of $\binom{n}{n/2}/4$. Still, even a height-1 schedule for this instance exists, as achieved by just alternating between any set and its complement. Thus, it is inevitable to take into account the structure of the set system rather than just the values of λ, which seems to be challenging in general.

To establish our result, we consider a slightly different setting than CBGT, which is also of independent interest: In BGT terms, upon cutting, the height of a bamboo is not reduced *to* 0 but *by* 1, possibly below 0. We call the supremum of all *absolute* heights (positive or negative) any bamboo ever reaches the *discrepancy* $d(\pi)$ of the corresponding schedule π. Later, we give a formal definition, and show that $d(\pi) < 1$ implies $h(\pi) < 2$ for all schedules π. (It is also not difficult to see that our result implies a height bound of 2 in the setting where one may cut by at most 1 and never below 0; see, e.g., [36, Section 4] for this setting.)

For this variant, our normalization (i.e., that g lies in the matroid polytope) is necessary to obtain a schedule of finite discrepancy. Indeed, suppose that an infinite schedule π has finite $d(\pi)$. As g is rational, each $e \in E$ can only take finitely many distinct heights without its absolute height exceeding $d(\pi)$, and thus we may assume that π eventually cyclically repeats some finite sequence π'. Each $e \in E$ must be cut by exactly as much as it grows in π', as otherwise $d(\pi)$ would diverge to infinity. Thus, taking $\lambda(I)$ to be the fraction of time steps $t \in [|\pi'|]$ such that $\pi'(t) = I$ for all $I \in \mathcal{I}$, we have $\sum_{I\in\mathcal{I}} \lambda(I) = 1$ and $\sum_{I\in\mathcal{I}} \lambda(I)\mathbb{1}_I = g$, which by definition implies that g lies in the matroid polytope of $(E, \mathcal{I})$.

Note that, if one could cut according to some point in the matroid polytope, one would simply be able to remove precisely the just-grown height. Minimizing discrepancy can thus also be understood as the problem of rounding the fractional growth process to an integral sequence of cuts. In other words, the question is: *How well can a continuous combinatorial process be approximated through a discrete one?* (See [24] for stating a similar question in the context of the matroidal adversarial version.)

The crucial ingredient is the following. Fix some $e \in E$ and $T \in \mathbb{N}$ such that $Tg(e) \in \mathbb{N}$. Then the sets $B \subseteq \{1, \dots, T\}$ of time points at which e can be cut so as to fulfill the discrepancy requirement are precisely the bases of a matroid. Not only that, the T-dimensional vector whose entries are identical $g(e)$ turns out to be contained in the corresponding matroid polytope.

This allows us to present the first formulation of BGT-like problems in terms of matroid intersection, which may be of independent interest. To this end, we define two matroids on the ground set $E \times [T]$ (for some T such that $Tg(e) \in \mathbb{N}$ for all $e \in E$): The direct sum M_T of T copies of the input matroid, and the direct sum M_E of the $|E|$ matroids that we just described. By our normalization and the above ingredient, we get that the vector $y^\star \in \mathbb{R}^{E \times [T]}$ with $y_{(e,t)} = g(e)$ for all $(e, t) \in E \times [T]$ is contained in the *matroid-intersection* polytope of M_T and M_E. This allows us to use the well-known integrality of the matroid-intersection polytope [13] to show that M_T and M_E have a common basis, an infinite repetition of which corresponds to our desired infinite schedule with discrepancy less than 1. Note that the integrality of matroid-intersection polytope has been previously used in the context of other, quite different, scheduling problems where an independent set has to be selected to be scheduled at every time step [23].

Ideally, we would like to have a polynomial-time algorithm for computing any next cut of such schedule. While mathematically clean, our above proof does not yield that; indeed, it seems inherently pseudo-polynomial. To address this drawback, we give polynomial-time algorithms for certain classes of matroids in the full version of this paper. Specifically, we give polynomial-time algorithms for uniform, partition, graphic, and laminar matroids, in each case assuming a natural representation of the input matroid. (See Section 2.) We show how *tree schedules* [4] can be used to get height-2 schedules on uniform (and thus partition) matroids in polynomial time. Then, using the concept of *rainbow-circuit-free colorings* [15, 18, 19, 27], we give polynomial-time algorithms for graphic and laminar matroids achieving height 4. Note that the presentation of these results has been omitted from this version of the paper due to space constraints. Please see the full version.

One may wonder whether Theorem 3.1 extends beyond matroids. Certainly, there are non-matroid independence systems such as $(\{a, b, c\}, \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}\})$, for which a height of 2 is also achievable. Indeed, here b and c can simply be treated as a single element from the ground set. We, however, establish that there are set systems for which not even a constant height is achievable.

▶ **Theorem 4.4** ($\star$). *For every $k \in \mathbb{N}$ there is a CBGT instance with $n = 2^k - 1$ elements where every infinite valid schedule has height at least $\frac{1}{2} \log_2(n + 1) - \frac{1}{2}$.*

While the proof of this theorem requires algebraic methods, we also give a simpler construction (in the proof of Theorem 4.3) that uses first principles but loses constant factors. We show that there exists an instance $(E, \mathcal{I}, g)$ in which $g(e) = 1/2$ for all $e \in E$ such that, for every subfamily $\mathcal{I}' \subseteq \mathcal{I}$ of sufficiently small size (but even of size $\Omega(\log |E|)$), there exists an element $e \in E$ with $e \notin \bigcup_{I \in \mathcal{I}'} I$. Consequently, if $\mathcal{I}'$ constitutes the first cuts in a schedule, some bamboo is never cut and therefore reaches height $\Omega(\log |E|)$.

Finally, we show that this upper bound can be matched asymptotically, even with a polynomial-time algorithm (assuming access to a maximum-weight independent-set oracle over the set system). Assuming that $\mathcal{I}$ is given explicitly, a polynomial-time algorithm computing an $O(\log |E|)$-height schedule with high probability was given in [11]. Our algorithm answers the open question of whether there exists a deterministic polynomial-time algorithm.

Table 1 Overview of our results. When result follow from other, stronger results, this is marked with an arrow in the direction of the respective stronger result. For results marked with "full version", please see this table in the full version of the paper for the corresponding theorems.

	upper bound		lower bound
	polynomial-time	existence	
general set systems	$O(\log \lvert E\rvert)$ (Theorem 4.7)	$O(\log \lvert E\rvert)$ ([11], Theorem 4.2)	$\Omega(\log \lvert E\rvert)$ (Theorem 4.4)
general matroids	–	2 (Theorem 3.1)	2 (1-uniform example)
graphic matroids	4 (full version)	2 (↑)	2 (1-uniform example)
laminar matroids	4 (full version)	2 (↑)	2 (1-uniform example)
uniform matroids	2 (full version)	2 (←, ↑)	2 (1-uniform example)
partition matroids	2 (full version)	2 (←, ↑)	2 (1-uniform example)

▶ **Theorem 4.7.** *For any CBGT instance $(E, \mathcal{I}, g)$ with n elements there is an infinite valid schedule π of period $2n$ such that $h(\pi) \le O(\log n)$. Furthermore, this schedule can be computed by making at most $2n$ queries to a maximum-weight independent-set oracle for $\mathcal{I}$.*

In fact, our algorithm implements a type of schedule that we can also obtain through a probabilistic argument combined with a straightforward case distinction into fast and slow bamboos. This technique was already used in [11] for the aforementioned result. We obtain the polynomial-time algorithm by choosing weights for the maximum-weight independent-set oracle such that we greedily minimize a carefully chosen potential function at each step.

We summarize our results in Table 1.

1.2 Further Related Work

We start by discussing vanilla (i.e., 1-uniform) perpetual-scheduling problems. In the paper that introduced BGT [16], it was shown by using a density bound (even a suboptimal one [14]) for vanilla PS that an infinite schedule achieving height at most 2 always exists. The same bound can also be achieved by a simpler deadline-driven strategy [36]. Only very recently, it was disproved that the Greedy algorithm, which at any time cuts the tallest bamboo, achieves a height of at most 2, by showing a lower bound of approximately 2.07 [26]; an upper bound of 4 is known [36].

Another line of work [12, 20, 31, 40, 43] is concerned not with bounding the achievable height across all instances but with, given an instance, efficiently computing a schedule that approximates the lowest height across all schedules for that instance. Very recently, a PTAS was obtained [33]. For CBGT, note that, when the minimum sum of coefficients to generate g is precisely 1, the height of any schedule is at least 1. Therefore, our bounds also imply results of the aforementioned flavor for our combinatorial variant.

Variants in which the growth rates may vary over time have also been considered. In cup games [1, 5, 6], the growth rate is chosen online by an adversary, and it is known that Greedy achieves a discrepancy of at most $O(\log n)$ for a ground set of size n. Online proportional apportionment [9] can be interpreted as the variant of this in which one may only cut bamboos that have just grown. In chairperson assignment [42], the varying growth rates are known ahead of time, and it is known [42] that a discrepancy of less than 1 can be achieved; a weighted version has been considered recently [39]. We note that the result for the unweighted problem together with our observation on the relationship between discrepancy and height (Lemma 3.2) also implies that a height-2 schedule for vanilla BGT exists, a fact that seems to have been missed by the literature.

For vanilla PS, as introduced in the 1990s [22], the density bound of 5/6 was recently confirmed in a breakthrough paper by Kawamura [31]. A tight example (periods 2, 3, and $x \in \mathbb{N}$) and several weaker bounds [10, 14, 38] had previously been known. Very recently, deciding feasibility of PS instances was shown to be NP-hard [33]. Recent works also consider a finite version of vanilla PS [29, 30] as well as a covering version [40].

Combinatorial variants where the independence system is a k-uniform matroid have been considered for pinhweel scheduling in form of a problem called *windows scheduling* [2], where it has been shown (implicitly) that the density threshold approaches 1 as $k \to \infty$. It has also been considered for cup games, where the achievable backlog is known to be in $O(\log n) \cap \Omega(\log(n-k))$ [37]. General matroids, with the same type of normalization, have also been considered for cup games [24], where the upper bound of $O(\log n)$ can be recovered.

In so-called *polyamorous scheduling* [8, 17], (essentially) a special case of CBGT, the ground set is the set of edges of a graph, and the set of independent sets is the set of matchings. For normalization, the authors assume that the vector of growth rates is contained in the *fractional* matching polytope. This type of normalization is completely identical to our normalization on bipartite graphs (due to the integrality of said polytope) and identical up to a factor of 3/2 on general graphs. It is known that a height of 4 is achievable, and a height of 5.24 in polynomial time [8]. The authors also show approximation hardness and consider a pinwheel version of that problem.

A generalization of the perpetual scheduling problem to set systems was previously proposed in [11] under the name *Fair Hitting Sequence Problem*. In this problem, the input is a triple $(A, \mathcal{F}, g')$ where $(A, \mathcal{F})$ is a set system as for CBGT. However, the set system is not necessarily downward closed, and the growth rates are on the sets of $\mathcal{F}$. A schedule is an infinite sequence of elements $a \in A$, and all sets $F \in \mathcal{F}$ containing a have their height reset to 0 upon cutting a. If we set $E = \mathcal{F}$, $\mathcal{I} = \{\{F \in \mathcal{F} \mid a \in F\} \mid a \in A\}$, and $g = g'$, then any schedule for $(A, \mathcal{F}, g')$ corresponds to a schedule for $(E, \mathcal{I}, g)$ with the same height, and thus the models are equivalent (up to downward closedness and normalization). While we focus on the existence and computation of low-height schedules, the authors of [11] focus on computing an approximation of the optimal schedule. They show that the exact computation of an optimal schedule is NP-hard, and also obtain a randomized polynomial-time $O(\log |E|)$-approximation algorithm via the rounding of an LP. It is not too difficult to see that their result actually implies the existence of an $O(\log |E|)$-height schedule with our normalization assumption. In Subsection 4.1, we give an alternative proof of the existence of a $O(\log |E|)$-height schedule and in Subsection 4.3, we give a deterministic polynomial-time algorithm which computes such a schedule. This answers an open question from [11].

1.3 Overview

In Section 2, we give definition and notation, and we also establish the relationship between CBGT and CPS. In Section 3, we prove the main result. In Section 4, we consider general independence systems. We state open problems in Section 5. Please find the omitted section on polynomial-time algorithms for special classes of matroids as well as proofs of results marked with $\star$ in the full version of the paper.

2 Preliminaries

We give the definitions used throughout the paper, repeating some of the definitions from the introduction. We first define $\mathbb{N} = \{1, 2, \dots\}$. We denote by $[n]$ the set $\{1, 2, \dots, n\}$ for any $n \in \mathbb{N}$. We denote by $\mathbb{Q}_{\geq 0}$ the set of nonnegative rational numbers.

Combinatorial Bamboo Garden Trimming. An instance of *Combinatorial Bamboo Garden Trimming* (CBGT) is a triple $(E, \mathcal{I}, g)$ where $(E, \mathcal{I})$ is an arbitrary set system and $g \in \mathbb{Q}^E_{\geq 0}$ is a growth rate vector. We require that g lies in the convex hull of the characteristic vectors of the independent sets in $\mathcal{I}$. That is, there exists a set of non-negative coefficients $\{\lambda(I) \mid I \in \mathcal{I}\}$ such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) = \sum_{I \in \mathcal{I}: e \in I} \lambda(I)$ for every $e \in E$.

A *schedule* π for a CBGT instance $(E, \mathcal{I}, g)$ is a finite or infinite sequence $\pi(1), \pi(2), \dots$ where $\pi(t) \subseteq E$ is the set of bamboos to cut on day t for every $t \in [|\pi|]$. If π is infinite, then we define $[|\pi|] = \mathbb{N}$. A schedule is *valid* if $\pi(t) \in \mathcal{I}$ for every $t \in [|\pi|]$ and otherwise it is *invalid*. The following definitions apply to *all* (finite/infinite valid/invalid) schedules.

For any $t \in [|\pi|]$ and $e \in E$, we denote by $h^\pi_t(e)$ the height of bamboo e on day t after e has grown but before $\pi(t)$ is cut. The *height* $h(\pi)$ of a schedule π is then

$$h(\pi) = \max_{e \in E} \sup_{t \in [|\pi|]} h^\pi_t(e)$$

or, in words, the maximum height reached by any bamboo during the execution of π. We can, equivalently, define the height of a schedule π as follows. The *recurrence time* $\gamma^\pi(e)$ of a bamboo $e \in E$ under a schedule π is the maximum number of days between consecutive schedulings of e according to π. That is, $\gamma^\pi(e) = \sup_{i, i' \in [|\pi|]} \{i' - i + 1 \mid e \notin \bigcup_{j=i}^{i'} \pi(j)\}$. The maximum height $h^\pi(e)$ reached by a bamboo $e \in E$ under π is then $h^\pi(e) = \gamma^\pi(e) \cdot g(e)$ and $h(\pi) = \max_{e \in E} h^\pi(e)$.

We also define the *discrepancy* $d(\pi)$ of a schedule π. Intuitively, the discrepancy is a measure of how closely the fraction of the time that e has been scheduled until day t matches the ideal fraction $t \cdot g(e)$. Define $A(\pi, e, t) = |\{1 \leq i \leq t \mid e \in \pi(i)\}|$ for all $t \in [|\pi|]$ as the number of occurrences of e among the first t sets of π. The discrepancy $d^\pi(e)$ of a bamboo $e \in E$ is then

$$d^\pi(e) = \sup_{t \in [|\pi|]} |t \cdot g(e) - A(\pi, e, t)|,$$

and we define the discrepancy $d(\pi)$ of π as $d(\pi) = \max_{e \in E} d^\pi(e)$. Note that $d^\pi(e) < 1$ if and only if $\lfloor tg(e) \rfloor \leq A(\pi, e, t) \leq \lceil tg(e) \rceil$ for $t \in [|\pi|]$.

Graph theory. Let $G = (V(G), E(G))$ be a graph. Given a subset $X \subseteq V(G)$, the *subgraph of G induced by X* is denoted by $G[X]$ and is obtained from G by removing all vertices in $V(G) \setminus X$ along with any edges incident to them. The *neighborhood* $N_G(v)$ of a vertex $v \in V(G)$ is the set of vertices adjacent to v in G.

Independence Systems and Matroids. We recall some basic definitions from matroid theory and refer the reader to [41] for further details.

We denote by $\mathbb{R}^E$ the set of $|E|$-dimensional real vectors indexed by $e \in E$. We denote by $\mathbf{1}^n$ the n-dimensional vector whose entries are all 1. We omit the superscript n when it is clear from context. The *convex hull* of a set $X \subseteq \mathbb{R}^n$ is denoted $\mathrm{conv}(X)$. The *incidence vector* of a subset $X \subseteq E$ is denoted by $\mathbb{1}_X$ and is defined as the $|E|$-dimensional vector such that $\mathbb{1}_X(e) = 1$ if $e \in X$ and otherwise $\mathbb{1}_X(e) = 0$ for all $e \in E$.

A set system M is a tuple $(E, \mathcal{I})$ where E is a set and $\mathcal{I}$ is any family of subsets of E. We call M an *independence system* if $\mathcal{I}$ is downward-closed ($Y \in \mathcal{I}, X \subseteq Y \Rightarrow X \in \mathcal{I}$) and $\emptyset \in \mathcal{I}$. The set E is called the *ground set* of M (also denoted $E(M)$) and $\mathcal{I}$ is called the set of *independent sets* of M (also denoted ($\mathcal{I}(M)$). We call M a *matroid* if M is an independence system and for all $X \subseteq E$, all inclusion-wise maximal independents subsets of X have the same size.

The *rank* $r_M(X)$ of a set $X \subseteq E$ is the maximum size of an independent subset of X. We omit the subscript M when M is clear from context. The *rank* of M itself is $r(E)$ and is also denoted $r(M)$. The maximal independent subsets of E are called *bases*.

The *matroid polytope* of a matroid M is the polytope $\mathcal{M} \subseteq \mathbb{R}^E$ such that $x \in \mathcal{M}$ if and only if

$$0 \leq \sum_{e \in X} x_e \leq r_M(X) \quad \forall X \subseteq E.$$

It is a well-known theorem of Edmonds [13] that $\mathcal{M} = \mathrm{conv}(\{\mathbb{1}_I \mid I \in \mathcal{I}(M)\})$, see e.g. [35, Theorem 13.12]. When $(E, \mathcal{I}, g)$ is an CBGT instance where $M = (E, \mathcal{I})$ is a matroid, the requirement that $g \in \mathrm{conv}(\{\mathbb{1}_I \mid I \in \mathcal{I}\})$ is thus equivalent to

$$0 \leq \sum_{e \in X} g(e) \leq r(X) \quad \forall X \subseteq E. \tag{1}$$

Let $M_1 = (E_1, \mathcal{I}_1), \ldots, M_k = (E_k, \mathcal{I}_k)$ be matroids on disjoint ground sets and let $E = E_1 \cup \cdots \cup E_k$. The *direct sum* $M_1 \oplus \cdots \oplus M_k$ is the matroid $M = (E, \mathcal{I})$, where $\mathcal{I} = \{I \subseteq E \mid I \cap E_i \in \mathcal{I}_i \text{ for } i \in [k]\}$.

Let G be a bipartite graph with $V(G) = S \cup T$. A subset $I \subseteq S$ is said to be *matchable* if there is a matching in G covering I. If $\mathcal{I}$ is the family of matchable sets, then $(S, \mathcal{I})$ is called *transversal matroid*. The *k-uniform matroid* with ground set E and rank k is defined by the independent sets $\{I \subseteq E \mid |I| \leq k\}$. A *partition matroid* is a direct sum of uniform matroids. The *graphic matroid* $(E, \mathcal{I})$ has an associated (undirected) graph G with edge set E, and its family of independent sets is $\mathcal{I} = \{I \subseteq E \mid I \text{ is the edge set of a forest in } G\}$. A family of subsets $\mathcal{L} \subseteq 2^E$ is *laminar* if for any $L, L' \in \mathcal{L}$, we have $L \subseteq L'$, $L' \subseteq L$, or $L \cap L' = \emptyset$. Given a laminar family $\mathcal{L}$ and a function $b \colon \mathcal{L} \to \mathbb{N}$, a matroid $(E, \mathcal{I})$ is *laminar* if its set system is described as $\mathcal{I} = \{I \subseteq E \mid |I \cap L| \leq b(L) \text{ for each } L \in \mathcal{L}\}$.

Algorithms. An *algorithm* $\mathcal{A}$ for a family $\mathcal{F}$ of CBGT instances takes as input an instance $(E, \mathcal{I}, g) \in \mathcal{F}$ and outputs a schedule π for $(E, \mathcal{I}, g)$ *one step at a time*. We measure the efficiency of $\mathcal{A}$ based on the number of operations used to compute the set of bamboos to at each step. We say that an algorithm $\mathcal{A}$ *implements* a schedule π if π is the sequence of sets obtained by running $\mathcal{A}$ indefinitely.

For our polynomial-time algorithms for k-uniform, graphic, and laminar matroids (which are presented in the full version of the paper) we assume that $\mathcal{I}$ is represented as just the integer k, the associated graph, and, respectively, the laminar family $\mathcal{L}$ and the upper bound

$b(L)$ for each $L \in \mathcal{L}$. It is with respect to this input representation that our algorithms are polynomial-time. In Section 4, where we give an algorithm implementing a height-$O(\log n)$ schedule for arbitrary independence systems[1], we assume that $\mathcal{I}$ is given implicitly via access to a *maximum-weight independent-set oracle*. Given non-negative weights $w(e)$ for $e \in E$, such an oracle can find a set $I \in \mathcal{I}$ with maximum total weight $\sum_{e\in I} w(e)$. Note that for several natural classes of set systems (e.g., matroids, intersections of two matroids, and matchings in general graphs) polynomial-time implementations of such an oracle are well known. Furthermore, the oracle is trivial to implement given $\mathcal{I}$ explicitly as, e.g., in [11].

2.1 Approximation Factor of Low-Height Schedules

The focus of this paper is on existence of schedules whose (absolute) height is low, and on algorithms for implementing such schedules. In this subsection, we show how one can turn such algorithms into *approximation algorithms*, i.e., algorithms whose output is guaranteed to be at most a factor of c away from the optimal schedule in terms of height.

In fact, all that is needed for an algorithm implementing a height-c schedule to be a c-approximation algorithm is that g is "maximally" scaled before it is given as input. For BGT, this observation reduces to the fact that a height-c schedule is a c-approximation schedule when g is such that $\sum_{e\in E} g(e) = 1$, which has been used in the literature. In general, consider the primal–dual pair of LPs displayed in Figure 1. It was observed in [11, Section 3] that $\Lambda(g)$, is a lower bound for the height $h(\pi^\star)$ of the optimal schedule. This gives the following, a proof of which we also give in the full version using the terminology of this paper.

▶ **Theorem 2.1** ($\star$). *Let $(E, \mathcal{I}, g)$ be a CBGT instance with $\Lambda(g) = 1$. The height $h(\pi^\star)$ of the optimal schedule $\pi^\star$ is at least* 1*.*

Note that, if $\Lambda(g) = 1$, then $g \in \operatorname{conv}(\{\mathbb{1}_I \mid I \in \mathcal{I}\})$. Indeed, the vector λ corresponding to the optimal primal solution is the set of coefficients of a convex combination $g' \in \operatorname{conv}(\{\mathbb{1}_I \mid I \in \mathcal{I}\})$ (as $\sum_{I\in\mathcal{I}} \lambda(I) = 1$) and $g' \geq g$ (by the constraints of the LP). We can transform λ such that all constraints (of the first type) are tight, without affecting $\sum_{I\in\mathcal{I}} \lambda(I)$, thus obtaining g as a convex combination of $\{\mathbb{1}_I \mid I \in \mathcal{I}\}$. Indeed, if there is an $e \in E$ with $g'(e) > g(e)$, then pick any $I \in \mathcal{I}$ containing e with $\lambda(I) > 0$ and move some mass from $\lambda(I)$ to $\lambda(I \setminus \{e\})$. Repeat until $g' = g$.

To obtain a c-approximation schedule from an algorithm computing a height-c schedule using Theorem 2.1, all we need to do is give $\hat{g} = g/\Lambda(g)$ to the algorithm instead of g. This guarantees that the obtained schedule π is a c-approximation with respect to $\hat{g}$ (as $h(\pi)/h(\pi^\star) \leq c/1$ with respect to $\hat{g}$), and thus also with respect to g as $h(\pi)/h(\pi^\star)$ is invariant under scaling of g.

Note that $\Lambda(g)$ can be computed in polynomial time with access to a maximum-weight independent-set oracle for $\mathcal{I}$. This follows directly from the fact that the separation problem for the dual (and thus the dual as a whole via the ellipsoid method) reduces to finding a maximum-weight independent set: Interpret a candidate solution $\{x_e\}_{e\in E}$ as weights and check if a maximum weight $I \in \mathcal{I}$ satisfies $\sum_{e\in I} x_e \leq 1$ using the oracle. If so, x is feasible. If not, return the violated dual constraint for this I.

While we find the assumption that $\mathcal{I}$ is downward-closed to be natural, we also consider the case of non-downward-closed $\mathcal{I}$ in Section 4 as this is formally required for our results to be comparable with [11]. An efficient $O(\log |E|)$-approximation algorithm can be obtained

[1] In fact, as we argue, the algorithm can be used for arbitrary (not necessarily downward-closed) set systems.

$$\Lambda(g) = \min \sum_{I\in\mathcal{I}} \lambda(I) \quad \text{s.t.} \sum_{I\in\mathcal{I}: e\in I} \lambda(I) \geq g(e) \quad \forall e\in E, \qquad \lambda(I)\geq 0 \quad \forall I\in\mathcal{I}.$$

(a) Primal.

$$\max \sum_{e\in E} x_e g(e) \quad \text{s.t.} \sum_{e\in I} x_e \leq 1 \quad \forall I\in\mathcal{I}, \qquad x_e \geq 0 \quad \forall e\in E.$$

(b) Dual.

Figure 1 Linear programs used to obtain approximation guarantees from height guarantees.

from our algorithm implementing a height-$O(\log|E|)$ schedule, which we present in Section 4, even for non-downward-closed set systems. For such set systems, it may not be that $g \in \mathrm{conv}(\{\mathbb{1}_I \mid I \in \mathcal{I}\})$, even when $\Lambda(g) = 1$. However, the proof of Theorem 2.1 does not assume that $\mathcal{I}$ is downward-closed, and the analysis of our height-$O(\log|E|)$ schedule only requires that there exists $g' \in \mathrm{conv}(\{\mathbb{1}_I \mid I \in \mathcal{I}\})$ such that $g' \geq g$, which certainly does exist when $\Lambda(g) = 1$ as we argued above.

2.2 Combinatorial Pinwheel Scheduling

An instance of *Combinatorial Pinwheel Scheduling* (CPS) is a triple $(E, \mathcal{I}, a)$ where $(E, \mathcal{I})$ is an arbitrary set system and $a \in \mathbb{N}^E$. We say that $(E, \mathcal{I}, a)$ is *schedulable* if there exists a valid infinite schedule π such that, for all $e \in E$, we have $e \in \bigcup_{i=t}^{t+a(e)-1} \pi(i)$ for all $t \in \mathbb{N}$. The *density* of $(E, \mathcal{I}, a)$ is the minimum ρ such that there exist $\lambda(I) \geq 0$ for all $I \in \mathcal{I}$ with $\sum_{i\in\mathcal{I}} \lambda(I) = \rho$ and $\sum_{I\in\mathcal{I}:\, e\in I} \lambda(I) \geq 1/a(e)$ for all $e \in E$. The *density threshold* $\rho^\star(\mathcal{F})$ for a family $\mathcal{F}$ of CPS instances is the largest value ρ such that any instance with density less than ρ is schedulable.

In this language, the recent confirmation of the pinwheel conjecture [31] shows that $\rho^\star(\mathcal{F}) = 5/6$ when $\mathcal{F} = \{(E, \mathcal{I}, a) \mid (E, \mathcal{I}) \text{ is a 1-uniform matroid}\}$. It follows from our results that $\rho^\star(\mathcal{F}) \geq 1/2$ for the much broader family $\mathcal{F} = \{(E, \mathcal{I}, a) \mid (E, \mathcal{I}) \text{ is a matroid}\}$. This is a consequence of the following straightforward observation.

Let $B = (E, \mathcal{I}, g)$ be a CBGT instance and let $P = (E, \mathcal{I}, a)$ be the CPS instance where $a(e) = \lfloor c/g(e) \rfloor$ for all $e \in E$ for some $c \in \mathbb{R}_{>0}$. Then P is schedulable if and only if there exists an infinite valid schedule π for B with $h(\pi) \leq c$.

Proof. Let π be a schedule for B. Then $h(\pi) \leq c$ iff $h^\pi(e) = \gamma_\pi(e) g(e) \leq c$ for all $e \in E$ iff $\gamma_\pi(e) \leq \lfloor c/g(e) \rfloor = a(e)$ for all $e \in E$ iff e occurs at least once in any period of $a(e)$ consecutive time steps in π for all $e \in E$. ◀

This, together with Theorem 3.1 (Section 3), implies a lower bound on the density threshold of $\{(E, \mathcal{I}, a) \mid (E, \mathcal{I}) \text{ is a matroid}\}$.

▶ **Corollary 2.2** ($\star$). *Let $P = (E, \mathcal{I}, a)$ be a CPS instance where $(E, \mathcal{I})$ is a matroid. If P has density at most $1/2$, then P is schedulable.*

Together with Theorem 4.3 (Subsection 4.2), Subsection 2.2 also implies an upper bound on the density threshold of $\{(E, \mathcal{I}, a) \mid (E, \mathcal{I}) \text{ is an arbitrary set system}\}$.

▶ **Corollary 2.3** ($\star$). *For any n, there exists a CPS instance $(E, \mathcal{I}, a)$ with $|E| \leq n$ and density $O(1/\log n)$ that is not schedulable.*

3 Existence of Height-2 Schedules on Matroids

In this section, we show Theorem 3.1, which we restate here for convenience.

▶ **Theorem 3.1.** *For any CBGT instance $(E, \mathcal{I}, g)$ where $(E, \mathcal{I})$ is a matroid, there exists an infinite valid schedule π for $(E, \mathcal{I}, g)$ with $h(\pi) < 2$.*

Throughout this section, fix a CBGT instance $(E, \mathcal{I}, g)$ where $M = (E, \mathcal{I})$ is a matroid. We assume that no $e \in E$ has $g(e) = 0$. This is without loss of generality as such e can simply be removed and the resulting submatroid considered instead. By (1), we can assume that

$$\sum_{e \in E} g(e) = r(M). \tag{2}$$

Indeed, if this does not hold, we modify g so that it does hold: Let $\lambda\colon \mathcal{I}(M) \to [0, 1]$ be such that $\sum_{I \in \mathcal{I}(M)} \lambda(M) = 1$ and $\sum_{I \in \mathcal{I}(M): e \in I} \lambda(I) = g(e)$ for all $e \in E$. If $\lambda(I) = 0$ for all non-bases $I \in \mathcal{I}(M)$, then

$$\sum_{e \in E} g(e) = \sum_{e \in E} \sum_{I \in \mathcal{I}(M)\colon e \in I} \lambda(I) = \sum_{I \in \mathcal{I}(M)} \lambda(I)|I| = r(M) \sum_{I \in \mathcal{I}(M)} \lambda(I) = r(M).$$

If $\lambda(I) > 0$ for some non-basis $I \in \mathcal{I}(M)$, then we can simply pick basis $I' \supset I$, increase $\lambda(I')$ by $\lambda(I)$ and set $\lambda(I) = 0$, iteratively until (2) holds. Clearly, since we did not decrease $g(e)$ for any $e \in E$, it suffices to show that the modified instance has a schedule of height less than 2.

In fact, we will show the stronger statement: $(E, \mathcal{I}, g)$ has a schedule π with $d(\pi) < 1$. We start by showing that this is indeed stronger.

▶ **Lemma 3.2.** *If π is a schedule with $d(\pi) < 1$, then $h(\pi) < 2$.*

Proof. Fix any $e \in E$. As $d(\pi) < 1$, we have

$$A(\pi, e, t) \in \{\lfloor g(e)t \rfloor, \lceil g(e)t \rceil\}$$

for all $t \in [|\pi|]$. Let $1 \le i < i'$. If $i' - i + 1 \ge 2/g(e)$, then

$$A(\pi, e, i') \ge \left\lfloor g(e) \left(i + \frac{2}{g(e)} - 1 \right) \right\rfloor = \lfloor g(e)(i-1) \rfloor + 2 > \lceil g(e)(i-1) \rceil \ge A(\pi, e, i-1)$$

and thus there must be some $j \in [i, i']$ such that $e \in \pi(j)$. Therefore,

$$h^\pi(e) = g(e)\gamma_\pi(e) = g(e) \sup_{1 \le i < i'} \left\{ i' - i + 1 \;\middle|\; e \notin \cup_{j=i}^{i'} \pi(j) \right\} < g(e) \frac{2}{g(e)} = 2.$$

As this holds for all $e \in E$, we have $h(\pi) = \max_{e \in E} h^\pi(e) < 2$, as desired. ◀

We remark that the converse of Lemma 3.2 does not hold. For example, for the BGT instance $(\{a, b\}, g)$ with $g(a) = 0.9$ and $g(b) = 0.1$, the schedule $\pi = ababab...$ has $h(\pi) = 1.8$, but $d(\pi)$ is unbounded.

We continue by showing that, for an appropriately chosen T, a finite schedule π with $|\pi| = T$ and $d(\pi) < 1$ can be repeated indefinitely to obtain an infinite schedule π^∞ with $d(\pi^\infty) < 1$.

▶ **Lemma 3.3.** *Let $(E, \mathcal{I}, g)$ be a CBGT instance. Let T be such that $Tg(e)$ is an integer for all $e \in E$ and let π be a schedule with $d(\pi) < 1$ and $|\pi| = T$. Then the infinite repetition π^∞ of π satisfies $d(\pi^\infty) < 1$.*

Proof. As $d(\pi) < 1$ and $Tg(e)$ is an integer, we have

$$|Tg(e) - A(\pi, e, T)| < 1 \Leftrightarrow Tg(e) - A(\pi, e, T) = 0.$$

Thus $|T'g(e) - A(\pi, e, T')| = |(T' \bmod T)g(e) - A(\pi, e, T' \bmod T)|$ for any $T' \in \mathbb{N}$, which implies $d(\pi^\infty) = d(\pi) < 1$. ◀

From now on, we fix the time horizon T as the least common multiple of the denominators of $\{g(e) \mid e \in E\}$. Note that for that choice of T, we have that $Tg(e)$ is an integer for all $e \in E$.

We aim to construct a finite schedule π for $(E, \mathcal{I}, g)$ with length T and discrepancy $d(\pi) < 1$. Our approach identifies schedules of length T with subsets of of the ground set $E \times [T]$. Specifically, any subset $S \subseteq E \times [T]$ corresponds to a schedule π where $e \in \pi(t)$ if and only if $(e, t) \in S$. Since an arbitrary subset $S \subseteq E \times [T]$ does not necessarily yield a valid schedule with low discrepancy, we enforce these constraints by defining two matroids, M_T and M_E, over the common ground set $E \times [T]$. We enforce validity by ensuring that the selection at every time step is an independent set in M; formally, we consider the the direct sum $M_T = \bigoplus_{t=1}^{T} M_t$ where each M_t is a copy of M on the ground set $E \times \{t\}$. The second matroid M_E is constructed to enforce the requirement that $d(\pi) < 1$.

The following lemma follows immediately from the definition of M_T.

▶ **Lemma 3.4.** *Any basis $I \in \mathcal{I}(M_T)$ corresponds to a valid finite schedule π (possibly with $d(\pi) \geq 1$).*

The (more involved) definition of matroid M_E is given in the next subsection. It will be defined such that it satisfies the following lemma, which is also proven in the next subsection.

▶ **Lemma 3.5.** *Any basis $I \in \mathcal{I}(M_E)$ corresponds to a (possibly invalid) finite schedule π with $d(\pi) < 1$.*

In the last subsection of this section, we show the following.

▶ **Lemma 3.6.** *M_E and M_T have a common basis.*

Assuming Lemmas 3.5 and 3.6, we now give a proof of the main theorem of this section.

Proof of Theorem 3.1. By Lemma 3.6, let $I \in \mathcal{I}(M_E) \cap \mathcal{I}(M_T)$ be a common basis of M_E and M_T and let π be the corresponding schedule. Lemmas 3.4 and 3.5 imply that π is a valid schedule and that $d(\pi) < 1$. By Lemma 3.3, $d(\pi^\infty) < 1$, where π^∞ is the infinite repetition of π. Finally, Lemma 3.2 allows us to conclude that $h(\pi^\infty) < 2$. ◀

3.1 Construction of M_E and Proof of Lemma 3.5

For each $e \in E$, we define a matroid M_e with ground set $\{e\} \times [T]$. This choice of taking $\{e\} \times [T]$ rather than just $[T]$ allow us to take a direct sum. We then define $M_E = \bigoplus_{e \in E} M_e$.

For the remainder of this subsection (except the proof of Lemma 3.5 at the end), let $e \in E$ be fixed. As e is fixed, we may simply consider $[T]$ to be the ground set of M_e instead of $\{e\} \times [T]$.

The intuition is as follows. We want a basis in M_e to correspond to a schedule π with $d^\pi(e) < 1$. If $d^\pi(e) < 1$, then the i-th cut of e in π must occur within a specific interval of time steps t: If the cut happens too early, then $A(\pi, e, t)$ exceeds $\lceil tg(e) \rceil$; if it happens too

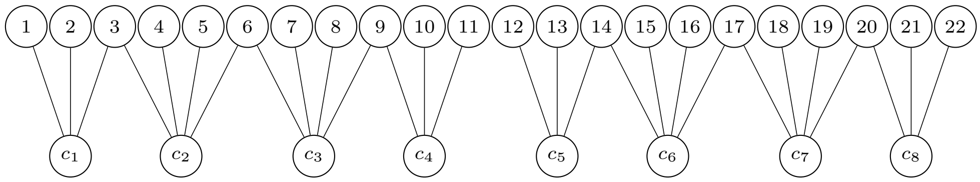

Figure 2 G_e for $g(e) = 4/11$ and $T = 22$.

late, then $A(\pi, e, t)$ falls behind $\lfloor tg(e) \rfloor$. We show that the interval C_i as defined below is exactly the interval of time steps in which the i-th cut of e must occur in order for $d^\pi(e) < 1$ to hold.

For $i \in \mathbb{N}$, define

$$C_i = [\min\{t \in \mathbb{N} \mid \lceil tg(e) \rceil = i\}, \min\{t \in \mathbb{N} \mid \lfloor tg(e) \rfloor = i\}] \cap \mathbb{N}. \tag{3}$$

Note that C_i is well defined as $0 < g(e) \leq 1$, and thus $\lfloor tg(e) \rfloor = i$ and $\lceil tg(e) \rceil = i$ have integer solutions for t for any $i \in \mathbb{N}$. Equivalently, we could define

$$C_i = \left[\left\lfloor \frac{i-1}{g(e)} \right\rfloor + 1, \left\lceil \frac{i}{g(e)} \right\rceil\right] \cap \mathbb{N}$$

for all $i \in \mathbb{N}$, which is straightforward to derive from (3).

We define G_e as the bipartite graph with $V(G_e) = [T] \cup \{c_1, c_2, \ldots, c_{Tg(e)}\}$ and $N_{G_e}(c_i) = C_i$ for $i \in [Tg(e)]$. The graph G_e for $T = 22$ and $g(e) = 4/11$ is illustrated in Figure 2. The set of independent sets $\mathcal{I}(M_e)$ of M_e is

$$\mathcal{I}(M_e) = \{I \subseteq [T] \mid I \text{ is covered by some matching in } G_e\}.$$

It is well-known that $M_e = ([T], \mathcal{I}(M_e))$ is a (transversal) matroid. To show that the bases of M_e correspond to schedules π with $d^\pi(e) < 1$, we first need an auxiliary lemma about the C_i.

▶ **Lemma 3.7.** *If $i \in \mathbb{N}$ is such that $i/g(e)$ is not an integer, then*

$$\min C_i < \max C_i = \min C_{i+1},$$

and if $i/g(e)$ is an integer, then $\max C_i + 1 = \min C_{i+1}$.

Proof. Let $i \in \mathbb{N}$. If $i/g(e)$ is an not an integer, then, as $g(e) \leq 1$,

$$\min C_i = \left\lfloor \frac{i-1}{g(e)} \right\rfloor + 1 \leq \left\lfloor \frac{i-1}{g(e)} + \frac{1}{g(e)} - 1 \right\rfloor + 1 = \left\lfloor \frac{i}{g(e)} \right\rfloor < \left\lceil \frac{i}{g(e)} \right\rceil = \max C_i$$

and

$$\max C_i = \left\lceil \frac{i}{g(e)} \right\rceil = \left\lfloor \frac{i+1-1}{g(e)} \right\rfloor + 1 = \min C_{i+1}.$$

Now, if $i/g(e)$ is an integer, then

$$\max C_i + 1 = \frac{i}{g(e)} + 1 = \frac{i+1-1}{g(e)} + 1 = \min C_{i+1},$$

which completes the proof. ◀

We next show two lemmas about the matroid M_e.

▶ **Lemma 3.8.** $r(M_e) = Tg(e)$.

Proof. Let $I \in \mathcal{I}(M_e)$. Note that $|I| > Tg(e)$ is not possible as I must be covered by some matching in G_e, and there are only $Tg(e)$ vertices $c_1, c_2, \ldots, c_{Tg(e)}$ to match I to. Furthermore, $\{\{\min C_I, c_i\} \mid i \in [Tg(e)]\}$ is a matching in G_e of size $Tg(e)$ as $\min C_i$ and $\min C_j$ are distinct for distinct i, j by Lemma 3.7. ◀

▶ **Lemma 3.9.** *If $I \in \mathcal{I}(M_e)$ is a basis and π is any schedule such that $e \in \pi(t)$ exactly when $t \in I$, then $d^\pi(e) < 1$.*

Proof. Let $k = Tg(e)$ and let $I \in \mathcal{I}(M_e)$ be a basis of M_e, which by Lemma 3.8 satisfies $|I| = k$. Let Y be a matching in G_e covering I, and let π be any schedule such that $e \in \pi(t)$ exactly when $t \in I$.

To show that $d^\pi(e) < 1$, we must show that

$$\lfloor tg(e) \rfloor \leq A(\pi, e, t) \leq \lceil tg(e) \rceil.$$

for all $t \in [T]$. To this end, fix any $t \in [T]$.

Let $I = \{t_1, t_2, \ldots, t_k\}$ be such that $t_1 < t_2 < \cdots < t_k$. For every $i \in [k]$, there must be some $j \in [k]$ such that $t_j c_i \in Y$ as $|I| = k$. We claim that $j = i$. Indeed, if this is not the case, then there is pair $i, j \in [k]$ with $i < j$ such that $t_j c_i \in Y$. But by Lemma 3.7, the intervals $C_{i+1}, C_{i+2}, \ldots, C_k$ do not contain any element less than t_j. This implies that $t_1, t_2, \ldots, t_j$ must be matched to $c_1, c_2, \ldots, c_i$, which is impossible for $i < j$. Hence, $t_i c_i \in Y$ holds for all $i \in [k]$.

We now show $A(\pi, e, t) \geq \lfloor tg(e) \rfloor$. Let $i = \lfloor tg(e) \rfloor$. As $t_i c_i \in Y$, we have $t_i \in N(c_i) = C_i$. Thus, the i-th occurrence of e in π is at index $t_i \leq \max C_i = \min\{t' \mid \lfloor t'g(e) \rfloor = i\} \leq t$, and thus $A(\pi, e, t) \geq i = \lfloor tg(e) \rfloor$.

We now show $A(\pi, e, t) \leq \lceil tg(e) \rceil$. Let $i = \lceil tg(e) \rceil$. If $i = k$, then we are done as π contains k occurrences of e in total. Thus, we assume $i \leq k - 1$. As $t_{i+1} c_{i+1} \in Y$, we have that $t_{i+1} \in N(c_{i+1}) = C_{i+1}$. Therefore, the $(i+1)$-th occurrence of e in π is at index $t_{i+1} \geq \min C_{i+1} = \min\{t' \mid \lceil t'g(e) \rceil = i + 1\} > t$, and thus $A(\pi, e, t) < i + 1$, which is equivalent to $A(\pi, e, t) \leq i = \lceil tg(e) \rceil$. ◀

We now give a proof of Lemma 3.5.

Proof of Lemma 3.5. Let $I \in \mathcal{I}(M_E)$ be a basis of M_E and let π be the corresponding schedule. As $M_E = \bigoplus_{e \in E} M_e$, the set $I \cap E(M_e)$ is a basis in M_e for all $e \in E$. As $e \in \pi(t)$ exactly when $(e, t) \in I \cap E(M_e)$, we have $d^\pi(e) < 1$ for all $e \in E$ by Lemma 3.9. Thus, $d(\pi) = \max_{e \in E} d^\pi(e) < 1$. ◀

3.2 Proof of Lemma 3.6

We start by observing that M_E and M_T have the same rank as

$$r(M_E) = \sum_{e \in E} r(M_e) = \sum_{e \in E} Tg(e) = T \cdot r(M) = \sum_{t=1}^{T} r(M_t) = r(M_T),$$

where we used the definition of M_E and M_T as direct sums for the first and last equalities, Lemma 3.8 for the second one, and our assumption that $\sum_{e \in E} g(e) = r(M)$ for the third.

Now, consider the *matroid intersection polytope* $\mathcal{P} \subset \mathbb{R}^{E\times[T]}$ of M_E and M_T where $y \in \mathcal{P}$ if and only if

$$0 \le \sum_{(e,t)\in X} y_{e,t} \le r_{M_T}(X) \quad \forall X \subseteq E \times [T] \tag{4}$$

$$0 \le \sum_{(e,t)\in X} y_{e,t} \le r_{M_E}(X) \quad \forall X \subseteq E \times [T]. \tag{5}$$

It is a well-known theorem of Edmonds [13] that $\mathcal{P}$ has integer vertices. Let $y^\star$ be such that $y^\star_{e,t} = g(e)$ for all $e \in E, t \in [T]$. We show that $y^\star \in \mathcal{P}$.

▶ **Lemma 3.10.** *$y^\star$ satisfies* (4)*.*

Proof. For each $t \in [T]$, the set of points $y \in \mathbb{R}^{E\times\{t\}}$ such that

$$0 \le \sum_{e\in Y} y_{e,t} \le r_{M_t}(Y \times \{t\}) \quad \forall Y \subseteq E$$

is exactly $\mathrm{conv}(\{\mathbb{1}_I \mid I \in \mathcal{I}(M_t)\})$ as M_t is a matroid. Thus, the above holds for $y^\star_{\cdot,t} = g$ for any $t \in [T]$ by the assumption that $g \in \mathrm{conv}(\{\mathbb{1}_I \mid I \in \mathcal{I}\})$. Thus, for any $X \subseteq E \times [T]$ we have that

$$0 \le \sum_{(e,t)\in X} y^\star_{e,t} = \sum_{t=1}^{T} \sum_{(e,t)\in X} y^\star_{e,t} \le \sum_{t=1}^{T} r_{M_t}(X \cap E(M_t)) = r_{M_T}(X).$$

This completes the proof. ◀

▶ **Lemma 3.11.** *$y^\star$ satisfies* (5)*.*

Proof. We show that $y^\star_{e,\cdot} = \mathbf{1}g(e)$ is a convex combination of $\{\mathbb{1}_I \mid \mathcal{I}(M_e)\}$ for any $e \in E$. If this holds, then as M_e is a matroid, it follows that $y^\star$ satisfies

$$0 \le \sum_{t\in Z} y^\star_{e,t} \le r_{M_e}(\{e\} \times Z) \quad \forall Z \subseteq [T]$$

for all $e \in E$, from which we get

$$0 \le \sum_{(e,t)\in X} y^\star_{e,t} = \sum_{e'\in E} \sum_{(e',t)\in X} y^\star_{e',t} \le \sum_{e\in E} r_{M_e}(X \cap E(M_e)) = r_{M_E}(X).$$

Fix any $e \in E$. We now proceed to show that $y^\star_{e,\cdot} = \mathbf{1}g(e) \in \mathrm{conv}(\{\mathbb{1}_I \mid \mathcal{I}(M_e)\})$ by showing that there exists a $\lambda : \mathcal{I}(M_e) \to [0,1]$ with $\sum_{I\in\mathcal{I}(M_e)} \lambda(I) = 1$ and $\sum_{I\in\mathcal{I}(M_e)} \lambda(I) \cdot \mathbb{1}_I = \mathbf{1}g(e)$.

Towards this, consider the *fractional matching polytope* $\mathcal{P}' \subset \mathbb{R}^{E(G_e)}$ of G_e (the graph through which we defined M_e) such that $x \in \mathcal{P}'$ iff

$$\begin{aligned} \sum_{uv\in E(G_e)} x_{uv} &\le 1 \quad \forall v \in V(G_e) \\ x_{uv} &\ge 0 \quad \forall uv \in E(G_e). \end{aligned}$$

Define

$$x_{c_it} = \begin{cases} tg(e) - i + 1 & \text{if } t = \min C_i \\ i - (t-1)g(e) & \text{if } t = \max C_i \\ g(e) & \text{otherwise} \end{cases}$$

for all $c_it \in E(G_e)$.

The proof of the following claim is a case distinction. It can be found in the full version of the paper.

▷ Claim 3.12 (⋆). We have $\sum_{c_i t \in E(G_e)} x_{c_i t} = g(e)$ for all $t \in [T]$, and $x \in \mathcal{P}'$.

Let $\mathcal{M}$ denote the set of all matchings in G_e. It is well-known that $\mathcal{P}'$ has integer vertices as G_e is bipartite (e.g., [35, Theorem 11.4]). Thus, every vertex of $\mathcal{P}'$ is the incidence vector $\mathbb{1}_A$ of some $A \in \mathcal{M}$. By this and as $x \in \mathcal{P}'$ (by Claim 3.12), x is a convex combination of $\{\mathbb{1}_A \mid A \in \mathcal{M}\}$. Let $\lambda' : \mathcal{M} \to [0,1]$ with $\sum_{A\in\mathcal{M}} \lambda'(M) = 1$ such that

$$\sum_{A\in\mathcal{M}} \lambda'(A) \cdot \mathbb{1}_A = x. \tag{6}$$

Now, for all $A \in \mathcal{M}$, define $T(A) = \{t \in [T] \mid t \text{ covered by } A\}$ as the set of endpoints of A in $[T]$. Define $\lambda : \mathcal{I}(M_e) \to [0,1]$ by setting

$$\lambda(I) = \sum_{A\in\mathcal{M}:\, T(A)=I} \lambda'(A)$$

for all $I \in \mathcal{I}(M_e)$. Then,

$$\sum_{I\in\mathcal{I}(M_e)} \lambda(I)\mathbb{1}_I = \sum_{I\in\mathcal{I}(M_e)} \left(\sum_{A\in\mathcal{M}:T(A)=I} \lambda'(A) \right) \mathbb{1}_I = \sum_{A\in\mathcal{M}} \lambda'(A)\mathbb{1}_{T(A)} = g(e)\mathbf{1},$$

where the first equality follow from the definition of $\lambda(I)$, the second from the fact $T(A) \in \mathcal{I}(M_e)$ for any matching $A \in \mathcal{M}$, and the last by Claim 3.12 and (6). Thus, we have expressed $g(e)\mathbf{1}^T$ as a convex combination of independent sets of M_e, which is what we wanted. ◀

We are now ready to prove Lemma 3.6.

Proof of Lemma 3.6. It follows from Lemma 3.10 and Lemma 3.11 that any $y \in \mathcal{P}$ which maximizes $\sum_{e\in E,t\in[T]} y_{e,t}$ has a sum of entries at least $\sum_{(e,t)\in E\times[T]} y^\star_{e,t} = T \cdot r(M) = r(M_T) = r(M_E)$. As $\mathcal{P}$ has integer vertices, which are binary by definition of $\mathcal{P}$, there exists a binary vector $y_{\text{bin}} \in \mathcal{P}$ with at least $r(M) \cdot T$ entries that are 1. By the definition of $\mathcal{P}$ and the fact that $r(M_E) = r(M_T) = r(M) \cdot T$, y_{bin} is the incidence vector of a common basis of M_E and M_T. ◀

4 General Set Systems

In this section, we give an efficient implementation of a height-$O(\log n)$ schedule for any set system, even those that are not independence systems. This improves upon a result from [11], which gives a randomized algorithm achieving the same guarantee (but only with high probability) and with running time polynomially dependent on $|\mathcal{I}|$[2]. Our algorithm assumes that $\mathcal{I}$ is given implicitly via access to a maximum-weight independent-set oracle, and makes at most $2|E|$ calls to this oracle. We remind the reader that a maximum-weight independent-set oracle takes weights $w_e \geq 0$ for $e \in E$ and returns a set $I \in \mathcal{I}$ with maximum $\sum_{e\in I} w_e$.

Throughout this section, we use $n = |E|$.

[2] Recall that an $O(\log n)$-approximation algorithm, which is the focus of [11], can be obtained from our algorithm by scaling g appropriately as described in Subsection 2.1.

4.1 Upper Bound

In this subsection, we show that a height-$O(\log n)$ schedule exists for any instance. We give a proof here (even though it was already shown in [11]) as warm-up for our efficient algorithm. We begin by introducing some notation. Let $c > 2$. We partition the elements into *slow* and *fast* sets according to the threshold $\tau_n = c\frac{\ln n}{n}$. Formally:

$$\begin{aligned} E_{\text{slow}} &= \{e \in E \mid g(e) \leq \tau_n\}, \\ E_{\text{fast}} &= \{e \in E \mid g(e) > \tau_n\}. \end{aligned}$$

The main idea is to devise two separate schedules, serving fast and slow elements separately. For the fast elements, we use the probabilistic method to show that a schedule generated by the coefficients of the convex combination suffices.

▶ **Lemma 4.1.** *There exists an infinite valid schedule π_{fast} such that the height of every fast element is bounded by $2c \ln n$ for all $t \geq 1$. That is,*

$$\max_{t \geq 1, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq 2c \ln n.$$

Proof. Let $\lambda \colon \mathcal{I} \to [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I} \colon e \in I} \lambda(I)$.

First, we prove the existence of a finite schedule σ of length n where the maximum height is bounded by $H = c \ln n$. We select a sequence of n independent sets $\sigma = (I_1, \ldots, I_n)$ randomly, where each I_t is chosen independently from $\mathcal{I}$ according to the probability distribution λ.

Consider a single element $e \in E$ and a time step $t \in [n]$. The height $h_t^\sigma(e)$ exceeds H only if e was not cut in the preceding $H/g(e)$ steps. Thus,

$$\begin{aligned} \mathbb{P}_\sigma[h_t^\sigma(e) > H] &\leq (1 - g(e))^{H/g(e)} \\ &\leq e^{-H} = e^{-c \ln n} = n^{-c}. \end{aligned}$$

By applying a union bound over all elements $e \in E$ and all time steps $t \in [n]$, the probability that any element exceeds height H at any timestep is

$$\mathbb{P}_\sigma \left[\max_{e \in E, t \in [n]} h_t^S(e) > H \right] \leq \sum_{e \in E} \sum_{t=1}^{n} \mathbb{P}_\sigma[h_t^S(e) > H] \leq n^2 \cdot n^{-c} = n^{2-c}.$$

Since $c > 2$, this probability is strictly less than 1. Therefore, there exists a sequence $\sigma = (I_1, \ldots, I_n)$ such that the height of any element within the schedule is at most H.

We construct the infinite schedule π_{fast} by concatenating infinite copies of σ. Consider a fast element $e \in E_{\text{fast}}$. By definition, $g(e) > \frac{c \ln n}{n} = H/n$, which implies $H/g(e) < n$. Since the height of e inside σ never exceeds H, the maximum gap between consecutive cuts of e in σ is at most $H/g(e) < n$. This guarantees that e is cut at least once in every copy of σ.

In the concatenated schedule π_{fast}, the maximum gap between two cuts is bounded by the gap across the boundary of two copies of σ, which is at most $2 \cdot \frac{H}{g(e)}$. Consequently, the maximum height of a fast element is bounded by

$$\max_{t \in \mathbb{N}, e \in E_{\text{fast}}} h_t^{\pi_{\text{fast}}}(e) \leq g(e) \cdot \left(\frac{2H}{g(e)} \right) = 2H = 2c \ln n. \quad ◀$$

For the slow elements, we serve them using a Round-Robin strategy. Since their growth rate is low, we only need to service them every $O(n)$ timesteps to keep the height below $O(\log n)$. We obtain the main result by interleaving the two schedules.

▶ **Theorem 4.2.** *For every $c > 2$, there exists a schedule π such that $h(\pi) \leq 3c \ln n$.*

Proof. We employ a weighted interleaving strategy. Let π_{fast} be the infinite schedule from Lemma 4.1. Let π_{slow} be a Round-Robin schedule of the n elements (i.e., it cuts $I_1, \ldots, I_n$ cyclically such that $e_1 \in I_1, \ldots, e_n \in I_n$). We construct a combined schedule π by using the fast schedule 2/3 of the time and the slow schedule the remaining third. Formally, for time steps $t \in \mathbb{N}$:

- If $t \neq 0 \pmod 3$, we execute the next move from π_{fast}.
- If $t = 0 \pmod 3$, we execute the next move from π_{slow}.

In the schedule π_{fast}, the height is bounded by $2c \ln n$. Thus, the height is bounded by:

$$\frac{3}{2}H = 3c \ln n.$$

In π, the schedule π_{slow} is executed every third step. Since π_{slow} has a period of n, every element is guaranteed to be cut at least once every $3n$ steps. For any $e \in E_{\text{slow}}$, the growth rate is $g(e) \leq \tau_n$. Thus, the maximum height for any $e \in E_{\text{slow}}$ is at most

$$3n \cdot \tau_n = 3n \cdot \frac{c \ln n}{n} = 3c \ln n.$$

We conclude that the schedule π has a height of at most $3c \ln n$. ◀

4.2 Lower Bound

We now show that the previous schedule is tight (up to constant factors). The main idea is that every instance where the growth rate is bounded from below and that has a schedule with low height must have a small covering by independent sets. Then, we try to construct instances with high growth rates that are hard to cover.

▶ **Theorem 4.3.** *For every $k \in \mathbb{N}$ there is a CBGT instance with $n = \binom{2k}{k}$ elements for which every infinite valid schedule has height at least $\frac{1}{4} \log_2 n$.*

Proof. Let the set of elements E be the collection of all subsets of $[2k]$ with size k; i.e.,

$$E = \{S \subseteq [2k] \mid |S| = k\}.$$

The collection of feasible cuts $\mathcal{I}$ consists of sets containing a fixed element from the universe $[2k]$. Specifically, for each $i \in [2k]$, we define:

$$I_i = \{S \in E \mid i \in S\}.$$

We let $\mathcal{I}$ be the downward closure of $\{I_1, I_2, \ldots, I_{2k}\}$.

We consider the growth vector given by the coefficients $\lambda(I_i) = \frac{1}{2k}$ for all $i \in [2k]$. These coefficients sum to $\sum_{i=1}^{2k} \lambda(I_i) = 1$, and for any element $S \in E$:

$$g(S) = \sum_{i \in [2k]\colon S \in I_i} \lambda(I_i) = \sum_{i \in S} \frac{1}{2k} = \frac{|S|}{2k} = \frac{k}{2k} = \frac{1}{2}.$$

We now show that for any infinite valid schedule, there exists an element $S^* \in E$ that is not cut during the first k time steps. Let the schedule select the cuts $I_{i_1}, I_{i_2}, \ldots, I_{i_k}$ in the first k steps. Let $J = \{i_1, \ldots, i_k\} \subseteq [2k]$ and note that $|J| \leq k$. Therefore, there exists a subset $S^* \subseteq [2k] \setminus J$ with $|S^*| = k$. Furthermore, this set has not been cut as $J \cap S^* = \emptyset$.

The height of S^* at time k is:

$$h_k(S^*) = k \cdot g(S^*) = \frac{k}{2}.$$

Using the bound $n = \binom{2k}{k} < 2^{2k}$, we have that $k > \frac{1}{2}\log_2 n$. Substituting this into the previous equation we obtain

$$h_k(S^*) > \frac{1}{2}\left(\frac{1}{2}\log_2 n\right) = \frac{1}{4}\log_2 n. \blacktriangleleft$$

A similar, albeit more involved, construction via algebraic methods can push the constant in the lower bound.

▶ **Theorem 4.4** ($\star$)**.** *For every $k \in \mathbb{N}$ there is a CBGT instance with $n = 2^k - 1$ elements where every infinite valid schedule has height at least $\frac{1}{2}\log_2(n+1) - \frac{1}{2}$.*

4.3 Efficient Implementation

We now show how to efficiently implement the schedule from Subsection 4.1. Note that we can find some $I \in \mathcal{I}$ with $e \in I$ by calling the maximum-weight independent-set oracle with $w_e = 1$ and $w_{e'} = 0$ for all $e' \neq e$. (If $\mathcal{I}$ is downward-closed, we can just take $I = \{e\}$.) Thus, computing π_{slow} requires only one oracle call per $e \in E_{\text{slow}}$.

▶ **Lemma 4.5.** *For any CBGT instance $(E, \mathcal{I}, g)$, the schedule π_{slow} can be computed by making n queries to a maximum-weight independent-set oracle for $\mathcal{I}$.*

We next describe how to implement an appropriate schedule for the fast elements. Given a height vector $h \in \mathbb{Q}^E_{\geq 0}$, we define the *potential* of h as $\Phi(h) = \sum_{e \in E_{\text{fast}}} \exp(h(e))$. Furthermore, if we are given a height vector $h \in \mathbb{Q}^E_{\geq 0}$ and an independent set $I \in \mathcal{I}$, we define *the potential of h after cutting I* as

$$\Phi(h, I) = \sum_{e \in E_{\text{fast}} \setminus I} \exp(h(e) + g(e)) + \sum_{e \in E_{\text{fast}} \cap I} \exp(g(e)).$$

Given a height vector $h \in \mathbb{Q}^E_{\geq 0}$, one can find an $I \in \mathcal{I}$ minimizing $\Phi(h, I)$ using the maximum-weight independent-set oracle. Indeed, observe that for any $I' \subseteq E$ and any $e \notin I'$ we have

$$\Phi(h, I' \cup \{e\}) = \begin{cases} \Phi(h, I') + \exp(g(e)) - \exp(g(e) + h(e)) & \text{if } e \in E_{\text{fast}}, \\ \Phi(h, I') & \text{otherwise.} \end{cases}$$

Thus, the problem of minimizing $\Phi(h, I)$ over $I \in \mathcal{I}$ is a maximum-weight independent-set problem when we set $w_e = \exp(g(e) + h(e)) - \exp(g(e)) \geq 0$ for $e \in E_{\text{fast}}$ and $w_e = 0$ for $e \in E \setminus E_{\text{fast}}$.

We show that the algorithm that greedily chooses a cut minimizing the potential at every time step (see Algorithm 1) is an appropriate schedule for the fast elements.

▶ **Lemma 4.6.** *For $n \geq 3$, the schedule computed by Algorithm 1 keeps the height of fast elements below $4 \ln n$ during the first n timesteps; i.e.,*

$$\max_{e \in E_{\text{fast}}, t \in [n]} h_t(e) \leq 4 \ln n.$$

Algorithm 1 Greedy Potential Scheduling.

Input: A CBGT instance $(E, \mathcal{I}, g)$.
Output: $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ a valid schedule of length n.

$h \leftarrow (0, \dots, 0)$
for $t = 1, \dots, n$ **do**
  Choose $I \in \mathcal{I}$ minimizing $\Phi(h, I)$.
  $\pi_t \leftarrow I$
  for $e \in E$ **do**
    If $e \notin I$, then $h(e) \leftarrow h(e) + g(e)$. Otherwise, $h(e) \leftarrow g(e)$.
  end for
end for

Proof. Let $\mathbf{0} = h_0, h_1, \dots, h_n \in \mathbb{Q}^E_{\geq 0}$ be the $n+1$ height vectors obtained during the execution of Algorithm 1. We define $\rho_n = 1 - \tau_n^3 < 1$. Let $\lambda \colon \mathcal{I} \to [0, 1]$ be such that $\sum_{I \in \mathcal{I}} \lambda(I) = 1$ and $g(e) \leq \sum_{I \in \mathcal{I} \colon e \in I} \lambda(I)$.

Note that $\Phi(h_{t+1}) = \min_{I \in \mathcal{I}} \Phi(h_t, I) \leq \mathbb{E}_I(\Phi(h_t, I))$, where we consider the random choice of an independent set $I \in \mathcal{I}$ according to the probability distribution λ. Thus,

$$\begin{aligned}
\Phi(h_{t+1}) &\leq \mathbb{E}_I(\Phi(h_t, I)) \\
&= \sum_{e \in E_{\text{fast}}} \exp(g(e)) \mathbb{P}_I[e \in I] + \sum_{e \in E_{\text{fast}}} \exp(h_t(e) + g(e)) \mathbb{P}_I[e \notin I] \\
&\leq \sum_{e \in E_{\text{fast}}} e + (1 - g(e)) \exp(g(e)) \exp(h_t(e)). \qquad (7)
\end{aligned}$$

Since $\exp(z) \leq 1 + z + z^2$ for $z \in [0, 1]$, we obtain

$$(1 - g(e)) \exp(g(e)) \leq (1 - g(e))(1 + g(e) + g(e)^2) = 1 - g(e)^3 \leq \rho_n. \qquad (8)$$

Putting (7) and (8) together, we obtain

$$\Phi(h_{t+1}) \leq en + \rho_n \Phi(h_t).$$

Applying the previous inequality repeatedly, for every $t \in [n]$ we obtain that

$$\Phi(h_t) \leq \Phi(h_0)\rho_n^n + en \sum_{i=1}^{n-1} \rho_n^i \leq n + \frac{en}{1 - \rho_n} \leq \frac{1 - \rho_n + e}{1 - \rho_n} n \leq \frac{(1 + e)n}{\tau_n^3}.$$

Using that $1 + e \leq (c \ln n)^3$ for $n \geq 3$, we conclude that for every $e \in E_{\text{fast}}$ and $t \in [n]$, it holds that

$$h_t(e) \leq \ln(\Phi(h_t)) \leq \ln\left(\frac{(1 + e)n^4}{(c \ln n)^3}\right) \leq 4 \ln n.$$

This completes the proof. ◀

Interleaving the schedule of Lemma 4.5 with the one of Lemma 4.6 we obtain the following.

▶ **Theorem 4.7.** *For any CBGT instance $(E, \mathcal{I}, g)$ with n elements there is an infinite valid schedule π of period $2n$ such that $h(\pi) \leq O(\log n)$. Furthermore, this schedule can be computed by making at most $2n$ queries to a maximum-weight independent-set oracle for $\mathcal{I}$.*

Proof. If $n \geq 3$ the interleaving of the schedules in Lemmas 4.5 and 4.6 gives the desired result. If $n < 3$, the schedule π_{slow} of Lemma 4.5 has a guarantee of 2. ◀

5 Open Problems

We conclude with some open problems.

Let us call a CBGT instance $(E, \mathcal{I}, g)$ a matroid instance if $(E, \mathcal{I})$ is a matroid. We have shown that any matroid CBGT instance has a schedule π with $h(\pi) < 2$. However, the implementation suggested by our proof uses time exponential in the number of bits required to represent g. Ideally, we would like an algorithm which is given access to $\mathcal{I}$ through some reasonable oracle (e.g. an indepence oracle) and which implements a schedule π with $h(\pi) < 2$ (or even $d(\pi) < 1$) in time polynomial in both $|E|$ and the number of bits required to represent g.

▶ **Open Problem 1.** *Does there exists a polynomial-time implementation of a height-2 schedule for matroid CBGT instances when access to $\mathcal{I}$ is given through some natural oracle?*

The notion of a *ℓ-system* was introduced by Jenkyns in the 1970s [28]. An independence systems $(E, \mathcal{I})$ is a ℓ-system if, for any $S \subseteq E$, the ratio of the smallest to the largest maximal independent subset of S is at most ℓ. Examples of ℓ-systems include the set systems obtained from the intersection of ℓ matroids [34]. Moreover, matroids are exactly the 1-systems. As we have shown that the family of matroid CBGT instances admit height-2 schedules, it is natural to ask the following.

▶ **Open Problem 2.** *Is there a function $f : \mathbb{N} \to \mathbb{N}$ such that all ℓ-system CBGT instances admit an $f(\ell)$-height schedule?*

We note that the lower bound presented in Theorem 4.3 is a $\Theta(|E|)$-system. Indeed, for the set system $(E, \mathcal{I})$ of Theorem 4.3, consider the subset

$$S = \{X \in E \mid 1 \in X \text{ and } 2 \notin X\} \subseteq E$$

and choose $X_1, X_2 \in E$ such that $X_1 \cap X_2 = \{2\}$. Note that $\{X_1, X_2\}$ and $\{X \in S \mid 1 \in X\}$ are the smallest and largest maximal independent sets respectively. Since their sizes are 2 and $\binom{2k-2}{k-1} \geq \frac{1}{4}\binom{2k}{k} = \frac{1}{4}|E|$ respectively, we conclude that $(E, \mathcal{I})$ is a $\Theta(|E|)$-system. This implies that if such an f exists, we must have $f(\ell) = \Omega(\log \ell)$.

References

1 Micah Adler, Petra Berenbrink, Tom Friedetzky, Leslie Ann Goldberg, Paul W. Goldberg, and Mike Paterson. A proportionate fair scheduling rule with good worst-case performance. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 101–108, 2003. `doi:10.1145/777412.777430`.

2 Amotz Bar-Noy and Richard E. Ladner. Windows scheduling problems for broadcast systems. *SIAM Journal on Computing*, 32(4):1091–1113, 2003. `doi:10.1137/S009753970240447X`.

3 Amotz Bar-Noy, Aviv Nisgav, and Boaz Patt-Shamir. Nearly optimal perfectly-periodic schedules. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 107–116, 2001. `doi:10.1145/383962.383998`.

4 Amotz Bar-Noy, Boaz Patt-Shamir, and Vladimir Dreizin. Efficient periodic scheduling by trees. In *Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 791–800, 2002. `doi:10.1109/INFCOM.2002.1019325`.

5 Michael A. Bender, Martin Farach-Colton, and William Kuszmaul. Achieving optimal backlog in multi-processor cup games. In *ACM Symposium on Theory of Computing (STOC)*, pages 1148–1157, 2019. `doi:10.1145/3313276.3316342`.

6 Michael A. Bender and William Kuszmaul. Randomized cup game algorithms against strong adversaries. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2059–2077. SIAM, 2021. doi:10.1137/1.9781611976465.123.

7 Kristóf Bérczi, Tamás Schwarcz, and Yutaro Yamaguchi. List coloring of two matroids through reduction to partition matroids. *SIAM Journal on Discrete Mathematics*, 35(3):2192–2209, 2021. doi:10.1137/20M1385615.

8 Yuriy Biktairov, Leszek Gasieniec, Wanchote Po Jiamjitrak, Namrata, Benjamin Smith, and Sebastian Wild. Simple approximation algorithms for polyamorous scheduling. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 290–314, 2025. doi:10.1137/1.9781611978315.23.

9 Javier Cembrano, Jose Correa, Svenja M. Griesbach, and Victor Verdugo. Online proportional apportionment. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4846–4860, 2026. doi:10.1137/1.9781611978971.174.

10 M. Y. Chan and F. Chin. Schedulers for larger classes of pinwheel instances. *Algorithmica*, 9:425–462, 1993. doi:10.1007/BF01187034.

11 Serafino Cicerone, Gabriele Di Stefano, Leszek Gasieniec, Tomasz Jurdzinski, Alfredo Navarra, Tomasz Radzik, and Grzegorz Stachowiak. Fair hitting sequence problem: Scheduling activities with varied frequency requirements. In *International Conference on Algorithms and Complexity (CIAC)*, pages 174–186, 2019. doi:10.1007/978-3-030-17402-6_15.

12 Federico Della Croce. An enhanced pinwheel algorithm for the bamboo garden trimming problem, 2020. Preprint, arXiv:2003.12460. doi:10.48550/arXiv.2003.12460.

13 Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In R. K. Guy, H. Hanani, N. Sauer, and J. Schönheim, editors, *Combinatorial Structures and Their Applications*, pages 69–87. Gordon and Breach, New York, 1970.

14 Peter C. Fishburn and J. C. Lagarias. Pinwheel scheduling: Achievable densities. *Algorithmica*, 34(1):14–38, 2002. doi:10.1007/S00453-002-0938-9.

15 Shinya Fujita, Colton Magnant, and Kenta Ozeki. Rainbow generalizations of ramsey theory: a survey. *Graphs and Combinatorics*, 26(1):1–30, 2010. doi:10.1007/s00373-010-0891-3.

16 Leszek Gasieniec, Tomasz Jurdzinski, Ralf Klasing, Christos Levcopoulos, Andrzej Lingas, Jie Min, and Tomasz Radzik. Perpetual maintenance of machines with different urgency requirements. *Journal of Computer and System Sciences*, 139:103476, 2024. doi:10.1016/j.jcss.2023.103476.

17 Leszek Gasieniec, Benjamin Smith, and Sebastian Wild. Polyamorous scheduling. In *International Conference on Fun with Algorithms (FUN)*, pages 15:1–15:18, 2024. doi:10.4230/LIPIcs.FUN.2024.15.

18 Adam Gouge, Dean Hoffman, Peter Johnson, Laura Nunley, and Luke Paben. Edge-colorings of K_n which forbid rainbow cycles. *Utilitas Mathematica*, 83, November 2010.

19 Dean Hoffman, Paul Horn, Peter Johnson, and Andrew Owens. On rainbow-cycle-forbidding edge colorings of finite graphs. *Graphs and Combinatorics*, 35(6):1585–1596, 2019. doi:10.1007/s00373-019-02102-6.

20 Felix Höhne and Rob van Stee. A 10/7-approximation for discrete bamboo garden trimming and continuous trimming on star graphs. In *International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 16:1–16:19, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.16.

21 Robert Holte, Aloysius Mok, Louis Rosier, Igor Tulchinsky, and Don Varvel. The pinwheel: A real-time scheduling problem. In *Proceedings of the 22nd Hawaii International Conference on System Science*, pages 693–702, Kailua-Kona, January 1989. doi:10.1109/HICSS.1989.48075.

22 Robert Holte, Louis Rosier, Igor Tulchinsky, and Donald Varvel. Pinwheel scheduling with two distinct numbers. *Theoretical Computer Science*, 100(1):105–135, 1992. doi:10.1016/0304-3975(92)90365-M.

23 Sungjin Im, Benjamin Moseley, Kirk Pruhs, and Manish Purohit. Matroid Coflow Scheduling. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132, pages 145:1–145:13, 2019. doi:10.4230/LIPIcs.ICALP.2019.145.

24 Sungjin Im, Benjamin Moseley, and Rudy Zhou. The matroid cup game. *Operations Research Letters*, 49(3):405–411, 2021. doi:10.1016/j.orl.2021.04.005.

25 Robert E. Jamison. Covering finite fields with cosets of subspaces. *Journal of Combinatorial Theory, Series A*, 22(3):253–266, 1977. doi:10.1016/0097-3165(77)90001-2.

26 Kalina Jasińska, John Kuszmaul, and Gyudong Lee. Strengths and limitations of greedy in cup games, 2026. Preprint, arXiv:2602.18610. doi:10.48550/arXiv.2602.18610.

27 Stanislav Jendrol', Jozef Miškuf, Roman Soták, and Erika Škrabul'áková. Rainbow faces in edge-colored plane graphs. *Journal of Graph Theory*, 62(1):84–99, 2009. doi:10.1002/jgt.20382.

28 T. A. Jenkyns. The efficacy of the "greedy" algorithm. In *Southeastern Conference on Combinatorics, Graph Theory, and Computing (SEICCGTC)*, pages 341–350, 1976.

29 Sotiris Kanellopoulos, Maria Kokkou, Euripides Markou, Aris Pagourtzis, and Christos Pergaminelis. Finite pinwheel scheduling: the k-visits problem. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 355–371, 2026. doi:10.1137/1.9781611978971.16.

30 Sotiris Kanellopoulos, Giorgos Mitropoulos, Christos Pergaminelis, and Thanos Tolias. Hardness, tractability and density thresholds of finite pinwheel scheduling variants. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2026. To appear.

31 Akitoshi Kawamura. Proof of the density threshold conjecture for pinwheel scheduling. In *ACM Symposium on Theory of Computing (STOC)*, pages 1816–1819, 2024. doi:10.1145/3618260.3649757.

32 Akitoshi Kawamura. Perpetual scheduling under frequency constraints. In Shin-ichi Minato, Takeaki Uno, Norihito Yasuda, Takashi Horiyama, Ken-ichi Kawarabayashi, Shigeru Yamashita, and Hirotaka Ono, editors, *Algorithmic Foundations for Social Advancement: Recent Progress on Theory and Practice*, pages 353–361. Springer, 2025. doi:10.1007/978-981-96-0668-9_23.

33 Robert Kleinberg and Ahan Mishra. Np-hardness and a ptas for the pinwheel problem, 2026. Preprint, arXiv:2604.13974. doi:10.48550/arXiv.2604.13974.

34 Bernhard Korte and Dirk Hausmann. An analysis of the greedy heuristic for independence systems. In B. Alspach, P. Hell, and D.J. Miller, editors, *Algorithmic Aspects of Combinatorics*, volume 2 of *Annals of Discrete Mathematics*, pages 65–74. Elsevier, 1978. doi:10.1016/S0167-5060(08)70322-4.

35 Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer Berlin Heidelberg, 6 edition, 2018. doi:10.1007/978-3-662-56039-6.

36 John Kuszmaul. Bamboo trimming revisited: Simple algorithms can do well too. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 411–417, 2022. doi:10.1145/3490148.3538580.

37 William Kuszmaul. Achieving optimal backlog in the vanilla multi-processor cup game. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1558–1577. SIAM, 2020. doi:10.1137/1.9781611975994.96.

38 Shun-Shii Lin and Kwei-Jay Lin. A pinwheel scheduler for three distinct numbers with a tight schedulability bound. *Algorithmica*, 19:411–426, 1997. doi:10.1007/PL00009181.

39 Siyue Liu and Victor Reis. Weighted chairman assignment and flow-time scheduling. In *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 98:1–98:15, 2026. doi:10.4230/LIPIcs.ITCS.2026.98.

40 Ahan Mishra. An optimal density bound for discretized point patrolling. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2846–2875, 2026. doi:10.1137/1.9781611978971.105.

41 James Oxley. *Matroid Theory*. Oxford University Press, February 2011. doi:10.1093/acprof:oso/9780198566946.001.0001.

42 Robert Tijdeman. The chairman assignment problem. *Discrete Mathematics*, 32(3):323–330, 1980. doi:10.1016/0012-365X(80)90269-1.

43 Martijn van Ee. A 12/7-approximation algorithm for the discrete bamboo garden trimming problem. *Operations Research Letters*, 49(5):645–649, 2021. doi:10.1016/j.orl.2021.07.001.