Better Algorithms and Hardness for Broadcast Scheduling via a Discrepancy Approach

Nikhil Bansal* Moses Charikar† Ravishankar Krishnaswamy‡ Shi Li§

Abstract

We study the broadcast scheduling problem with the objective of minimizing the average response time. There is a single server that can hold n pages of unit size, and multiple requests for these pages arrive over time. At each time slot the server can broadcast one page which satisfies all the outstanding requests for this page at that time. The goal is to find a schedule to minimize the average response time of the requests, i.e. the duration since a request arrives until it is satisfied.

We give an $\tilde{O}(\log^{1.5} n)$ approximation algorithm for the problem improving upon the previous $\tilde{O}(\log^2 n)$ approximation. We also show an $\Omega(\log^{1/2-\epsilon} n)$ hardness result, and an integrality gap of $\Omega(\log n)$ for the natural LP relaxation for the problem. Prior to our work, only NP-Hardness and a (tiny) constant integrality gap was known. These results are based on establishing a close connection to the discrepancy minimization problem for permutation set-systems. Specifically, our improved approximation is based on using recent algorithmic ideas developed for discrepancy minimization. Our integrality gap is obtained from the $\Omega(\log n)$-lower bound on the discrepancy of 3-permutations, while our hardness result is based on establishing the first hardness result for the discrepancy of ℓ-permutations.

1 Introduction

We consider the classic broadcast scheduling setting which has received a lot of attention in recent years. The problem we study is formalized as follows: there is a collection of pages $P = \{1, \ldots, n\}$. Pages are broadcast by a server in integer time slots in response to requests. At time t, the server receives $w_p(t)$ requests for page $p \in P$. We say that a request ρ for page p that arrives at time t is satisfied at time $c_p(t)$, if $c_p(t)$ is the first time after t when page p is transmitted by the server. The response time of the request ρ is defined to be the time that elapses from its arrival till the time it is satisfied, i.e. $c_p(t) - t$. We assume that the response time for any request is at least 1. The goal is to find a schedule for broadcasting pages to minimize the average response time, i.e. $(\sum_{t,p} w_p(t)(c_p(t) - t))/\sum_{t,p} w_p(t)$. Here, we study the offline problem, where the request sequence is known in advance to the scheduling algorithm.

The problem was shown to be NP-hard by Erlebach and Hall [19]. Most of the initial focus was in the the resource augmentation setting. These results compare the k-speed algorithm, which is allowed to broadcast k pages per time slot, against the performance of an optimal 1-speed algorithm. Kalyanasundaram et al. [24] gave the first $(1/\alpha)$-speed, $1/(1 - 2\alpha)$-approximation algorithm for any $\alpha \leq 1/3$. This guarantee was improved in sequence of papers [20, 19, 21] culminating in a $(1 + \epsilon)$-speed, $O(1/\epsilon)$ approximation for any $\epsilon > 0$ [3]. When no extra speed is allowed, the problem gets considerably harder. Note that repeatedly broadcasting the pages in the cyclic order $1, \ldots, n$ is a trivial $O(n)$ approximation. The first improvement over this was an $O(\sqrt{n})$ approximation due to [3]. Later, a significantly improved $O(\log^2 n)$ approximation algorithm was given by [4]. These results are all based on rounding a natural LP relaxation for the problem. The ideas and the framework introduced in [4] will play a key role in this paper.

The problem of minimizing the average response time has also been studied extensively in the online setting [24, 17, 18, 3, 30, 14, 15, 23, 5]. In particular, several $(1 + \epsilon)$-speed, $O(\text{poly}(1/\epsilon))$ are now known [23, 5], and it is also known that extra speed is necessary to obtain $n^{o(1)}$ competitive algorithms. In addition to average response time, various other measures such as maximum response time [7, 11] and throughput maximization [6, 21, 10] have also been studied in the broadcast setting.

Our Results. In this paper, we give an improved algorithm with approximation guarantee $O(\log^{1.5} n)$ for the offline problem. In particular, we show the following

*TU Eindhoven bansal@gmail.com. This research has been supported by the NWO grant 639.022.211.

†Princeton University moses@cs.princeton.edu. Supported by NSF grants CCF-0832797, CCF-0916218 and CCF-1218687.

‡Princeton University ravishan@cs.cmu.edu. Supported by a Simons postdoctoral fellowship.

§Princeton University shili@cs.princeton.edu. Supported by NSF grants CCF-0832797, CCF-0916218, CCF-1218687 and a Wallace Memorial Fellowship.

more general result.

THEOREM 1.1. *There is a polynomial time algorithm that finds a schedule with average response time* $3 \cdot OPT + O((\sqrt{\log n \cdot \log \log n}) \log n)$*, where OPT denotes the value of the average response time in the optimum schedule.*

While we only show the above result in this paper, using the idea in Lemma 11 in [4] gives the following parametrized theorem.

THEOREM 1.2. *Let* $\gamma > 0$ *be any arbitrary parameter. There is a polynomial time algorithm that finds a schedule with average response time* $(2+\gamma) \cdot OPT + O((\sqrt{\log_{1+\gamma} n \cdot \log \log n}) \log n)$*, where OPT is the value of the average response time in the optimum schedule.*

Setting $\gamma = \Theta(\log n)$ above implies the claimed approximation guarantee of $O(\log^{1.5} n)$. Note that the $\tilde{O}(\log^{1.5} n)$ term in Theorem 1.1 is additive. In particular, if OPT is large, say OPT $= \Omega(\log^{1.5+\epsilon} n)$ for some $\epsilon > 0$, then setting γ arbitrarily small, implies an approximation ratio arbitrarily close to 2.

The main idea behind our algorithm is to build on the approach of [4] using ideas similar to those used by Rothvoss in his recent breakthrough result [31] for bin-packing. Roughly speaking, the algorithm in [4] works as follows. It solves the natural LP relaxation for the problem, and reduces the problem of finding a good rounding for it, to finding a good integral solution to an auxiliary linear program. This auxiliary LP is then solved using $O(\log n)$ rounds of iterative rounding, where an $O(\log n)$ factor is lost at each round. Here, we follow the same framework, but instead show how to use a recent result of Lovett and Meka [27] (developed in the context of discrepancy minimization) so that the iterative step incurs only an $\tilde{O}(\sqrt{\log n})$ factor loss in each round. To this end, we exploit several structural properties of the auxiliary LP, and in particular show how they can be used to write a third LP on which the Lovett-Meka algorithm can be run.

We also complement the above result with several negative results. First, we substantially improve the best known integrality gap of 28/27 [3] for the natural LP relaxation (on which all the known results are based).

THEOREM 1.3. *The natural LP relaxation for the broadcast problem has an integrality gap of* $\Omega(\log n)$.

Even though the algorithmic approach in [4] (and our algorithm) has similarities to bin-packing, this result suggests that the broadcast problem is substantially harder than bin-packing for which even an additive +1 integrality gap has not been ruled out. Interestingly, Theorem 1.3 is based on establishing a new connection with the problem of minimizing the discrepancy of 3-permutations. In the 3-permutation problem, we are given 3-permutations π_1, π_2, π_3 of $[n]$. The discrepancy of $\Pi = (\pi_1, \pi_2, \pi_3)$ w.r.t a ± 1 coloring χ is the worst case discrepancy of all prefixes. That is, $\max_{i=1}^{3} \max_{k=1}^{n} \left| \sum_{j=1}^{k} \chi(\pi_{i,j}) \right|$, where $\pi_{i,j}$ is the j^{th} element in π_i. The goal of the problem is to find a coloring χ that minimizes the discrepancy. Recently Newman and Nikolov [28] showed a (tight) $\Omega(\log n)$ lower bound on the discrepancy of 3-permutations, resolving a long standing conjecture. Our integrality gap in Theorem 1.3 is obtained by combining this result with our connection to the discrepancy of 3-permutations.

Then, by generalizing the connection to the discrepancy of ℓ-permutations, we show the following hardness result (prior to this, only NP-hardness was known [19]).

THEOREM 1.4. *There is no* $O(\log^{1/2-\epsilon} n)$*-approximation algorithm for the problem of minimizing average response time, for any* $\epsilon > 0$*, unless* $\mathsf{NP} \subseteq \bigcup_{t>0} \mathsf{BPTIME}(2^{\log^t n})$.

Along the way to proving Theorem 1.4, we also give the first hardness result for the ℓ-permutation problem.

THEOREM 1.5. *There is no* $\alpha(\ell)$*-approximation algorithm for the* ℓ*-permutation problem, for any sufficiently large* ℓ *and some function* $\alpha(\ell) = \Omega(\ell^{1/2})$*, unless NP = RP.*

Actually, we prove a stronger theorem: it is hard to distinguish between the systems $\Pi = (\pi_1, \pi_2, \cdots, \pi_\ell)$ with discrepancy $O(1)$ and with *average discrepancy* $\Omega(\ell^{1/2})$. The average discrepancy of a coloring χ is $\frac{1}{\ell} \sum_{i=1}^{\ell} \max_{k=1}^{n} \left| \sum_{j=1}^{k} \chi(\pi_{i,j}) \right|$. It turns out that average discrepancy is the right notion which helps us show our hardness for broadcast scheduling. We emphasize that in the ℓ-permutation problem, ℓ is not a part of the input. Otherwise, by letting ℓ equal to the permutation size n (and using our results on completing partial permutations from Section 3.3), the $\Omega(\ell^{1/2})$-hardness result can be obtained from the hardness for set discrepancy established by Charikar, Newman and Nikolov [9]. We also remark that, in order to prove Theorem 1.4, we need to allow some weak dependence between ℓ and n. Specifically, we allow ℓ to be $\operatorname{poly}\log(n)$.

Organization. In section 2 we describe the framework of [4] and describe our improved algorithm. In sections 3 and 4, we establish hardness for discrepancy of permutations and the integrality gap and hardness for the broadcast problem.

2 Improved Algorithm for Broadcast Scheduling

Our rounding algorithm follows the same high-level approach as the BCS algorithm [4]. We first solve the natural LP relaxation $\mathsf{LP}_{\mathsf{bcast}}$ for the problem. Using a two-phase rounding scheme, we construct a *tentative schedule* (which may transmit multiple pages at the same time) which satisfies the following properties:

- The total response time for this schedule is at most $c = O(1)$ times the cost of the LP relaxation.
- The capacity constraints are satisfied approximately in the following sense. For any interval of time $(t, t']$, the total number of pages broadcast by the tentative schedule during $(t, t']$ is at most $t' - t + b$, for some fixed value b. We refer to this b as the *backlog* of the schedule.

The following lemma shows that a low backlog schedule can be converted to a good feasible schedule (see Lemma 7, [4] for a simple proof).

LEMMA 2.1. *Any tentative schedule with backlog b and average response time c can be transformed into a feasible schedule with average response time at most $c + b$.*

The LP Relaxation. The starting point is the following integer programming formulation for our problem[1]. For each page $p \in [n]$ and each time t', there is a variable $y_{pt'}$ which indicates whether page p was transmitted at time t'. We have another set of variables $x_{ptt'}$ s.t $t' > t$, which indicates whether a request for page p which arrives at time t is satisfied at t'. Let w_{pt} denote the total weight of requests for page p that arrive at time t. Recall that requests arrive at the end of time slots and can only be served by the next time slot.

$$\min \sum_p \sum_t \sum_{t'=t}^{T_{\max}+n} (t'-t) \cdot w_{pt} \cdot x_{ptt'} \tag{2.1}$$

$$\text{s.t.} \sum_p y_{pt} \leq 1 \qquad \forall t \tag{2.2}$$

$$\sum_{t'=t+1}^{T_{\max}+n} x_{ptt'} \geq 1 \qquad \forall p, t \tag{2.3}$$

$$x_{ptt'} \leq y_{pt'} \qquad \forall p, t, t' \geq t \tag{2.4}$$

$$x_{ptt'} \in \{0, 1\} \qquad \forall p, t, t' \tag{2.5}$$

$$y_{pt'} \in \{0, 1\} \qquad \forall p, t' \tag{2.6}$$

Here $T_{\max}$ denotes the last time when any request arrives. Observe that it suffices to define variables only until time $t = T_{\max} + n$ as all requests can be satisfied by transmitting pages $1, \ldots, n$ after time $T_{\max}$. Constraint (2.2) ensures that only one page is transmitted in each time, (2.3) ensures that each request must be satisfied, and (2.4) ensures that a request for page p can be satisfied at time t only if p is transmitted at time t. Finally, a request arriving at time t that is satisfied at time t' contributes $(t'-t)$ to the objective. Now consider the linear program obtained by relaxing the integrality constraints on $x_{ptt'}$ and y_{pt}.

Notation. Let (x^*, y^*) denote an optimal solution to $\mathsf{LP}_{\mathsf{bcast}}$. We can WLOG assume that there is a request for every page at each time step, by setting the weight $w_{pt} = 0$ if there is no request for page p at time t. For any page p and time t, let $\sigma(p, t)$ denote the fractional response time for a request $r(p, t)$ that arrives at time t. Formally, $\sigma(p, t) = \sum_{t' > t}(t' - t)x^*_{ptt'}$. Also, let $y^*(p, t_1, t_2) = \sum_{t=t_1}^{t_2} y^*_{pt}$ denote the cumulative broadcast of page p between times t_1 and t_2. Finally, given a time interval $[t_1, t_2]$, let $t_- = \arg\max_{t'} y^*(p, t', t_2) \geq 1$ (or is t_1 is no such t' exists). Define $\mathsf{head}(I) = [t_1, t_-]$ and $\mathsf{tail}(I) = [t_-, t_2]$. In words, $\mathsf{tail}(I)$ is the smallest suffix of I where there is one unit of broadcast of p, and $\mathsf{head}(I)$ is the corresponding prefix.

2.1 Phase 1: Getting a Block Structured Fractional Solution

Since the first phase follows exactly as in [4], and the novel component is our rounding in the second phase, we mainly focus on the second phase while only presenting the necessary aspects about the first. Indeed, the first phase of the algorithm of Bansal et al. [4] (henceforth called the BCS algorithm) outputs a feasible fractional solution to the following *auxiliary LP* defined over the following variables.

Blocks and Shifts. For each page p, the BCS algorithm divides the time horizon $[1, T_{\max} + n]$ into disjoint intervals called *blocks*, each of which has at most $O(\log T_{\max})$ cumulative fractional broadcast $y^*(p, B)$ of page p. Moreover, within each block $B = [t_1, t_2)$ for page p, the BCS algorithm identifies a set of possible *tentative schedules* for the page p within the block. Intuitively, these are the different *offset schedules* within the block. e.g., for a fixed offset $\alpha \in (0, 1]$, the corresponding offset schedule makes tentative broadcasts at those time slots $t_{i+\alpha}$ when the cumulative broadcast $y^*(p, t_1, t_{i+\alpha})$ first exceeds $i + \alpha$, for $i \in \mathbf{Z}_{\geq 0}$.

Auxiliary LP. In what follows, we denote a (block,offset) tuple (B, α) as a *shift* S. There is a variable X_S for each shift. For a block B, let $p(B)$ denote the associated page, and $S(B)$ denote the set of all shifts associated with B. Let $\mathcal{S}$ denote the set of all shifts. Finally, for shift S, let $p(S)$ denote the associated page, and $\mathbf{1}(S, t)$ be the indicator variable for whether shift

[1] As stated the LP size is polynomial in $T_{\max}$. However we can assume that the problem input size is at least $T_{\max}/n$, since if there is period of n consecutive timesteps when no page is requested, then we can split the problem into two instances.

$S \in \mathcal{S}$ makes a tentative broadcast at time t.

We have the following constraints: (i) each block B chooses exactly one shift from $S(B)$, and (ii) any time t can have at most one broadcast. Finally each shift has a cost $C(S)$ defined appropriately.

Bansal et al. [4] show the following results about these blocks and the auxiliary LP, which we compress into one lemma.

$$(\mathsf{LP}_{\mathsf{aux}}) \qquad \min \sum_{S \in \mathcal{S}} x_S C(S)$$

$$(2.7) \qquad \sum_{S \in S(B)} x_S = 1 \qquad \forall B \in \mathcal{B},$$

$$(2.8) \qquad \sum_{S \in \mathcal{S}} \mathbf{1}(S,t) x(S) \le 1 \qquad \forall t \in [T_{\max}],$$

$$(2.9) \qquad x_S \ge 0 \qquad \forall S \in \mathcal{S}.$$

Lemma 2.2. *After Phase I of [4], the constructed blocks, their shifts, and the auxiliary LP $\mathsf{LP}_{\mathsf{aux}}$ satisfy the following properties.*

(i) Any shift $S \in \mathcal{S}$ makes at most $O(\log T_{\max})$ tentative broadcasts, all within its corresponding block.
(ii) The different shifts in a block are "interleaving": between two successive transmissions of a shift $S' \in S(B)$, there is exactly one transmission made by any other shift $S \in S(B)$.
(iii) $\mathsf{LP}_{\mathsf{aux}}$ has a feasible solution of cost at most $3 \cdot OPT$.
(iv) For any integral assignment $X_S \in \{0,1\}$ which satisfies (2.7), the cost of the tentative schedule is exactly its objective value in $\mathsf{LP}_{\mathsf{aux}}$.

The proof is in Appendix A.1. Note that apriori $T_{\max}$ can be arbitrarily large (w.r.t n). However, by Lemma 10 in [4], we can assume that $T_{\max} = \mathrm{poly}(n)$ using a decomposition procedure. In what follows, we will still use $O(\log T_{\max})$ below to make the dependence on time horizon explicit for better clarity. Furthermore, the factor 3 in part (iii) of the lemma above can be reduced to $2+\gamma$ for any $\gamma > 0$, at the expense of increasing the number of broadcasts in a shift to $O(\log_{1+\gamma} T_{\max})$ (see Lemma 11 in [4]).

2.2 Phase 2: Rounding the Auxiliary LP

Let $\mathbf{x}^*$ denote an optimal solution to $\mathsf{LP}_{\mathsf{aux}}$. The remainder of this section focuses on rounding $\mathbf{x}^*$.

Theorem 2.1. *LP solution $\mathbf{x}^*$ to $\mathsf{LP}_{\mathsf{aux}}$ can be efficiently rounded to an integral solution $\mathbf{X} = \{X_S\}$ satisfying*

(i) $\sum_S X_S C(S) \le O(1) \sum_S x^*_S C(S)$,
(ii) $\sum_{S \in S(B)} X_S = 1$, for all $B \in \mathcal{B}$, and
(iii) For any interval $I = [t_1, t_2)$, $\sum_S X_S \mathbf{1}(S, I) \le (t_2 - t_1) + b$, for $b = O(\log^{3/2} T_{\max})$. In words, the total number of tentative broadcasts is at most $(t_2 - t_1) + b$ for any interval $[t_1, t_2)$.

Here, $\mathbf{1}(S, I) = \sum_{t_1 \le t < t_2} \mathbf{1}(S,t)$ denotes the total number of broadcasts that shift S makes in time interval $I = [t_1, t_2)$. We can then recover a schedule with average response time $O(1) \cdot \mathrm{OPT} + b$, by using property (iv) of Lemma 2.2 and Lemma 2.1.

2.2.1 Proof of Theorem 2.1: The Rounding Algorithm

Our rounding of $\mathsf{LP}_{\mathsf{aux}}$ will proceed in iterations. In each iteration, we will round a constant fraction of the variables x_S to integral values in $\{0,1\}$. Throughout the algorithm we will maintain the invariant that $\sum_{S \in S(B)} x_S = 1$ for each block B. This implies that once some variable x_S is set to 1, then $x_{S'} = 0$ for all $S' \in S(B) \setminus \{S\}$ where B is the block containing shift S. We call such a block B integrally assigned.

Algorithm Notation. Let $\mathbf{X}_k$ (boldfont) denote the fractional solution at the beginning of the k^{th} iteration, and $X_{S,k}$ denote the value of the LP variable corresponding to shift S in $\mathbf{X}_k$. Let $\mathcal{S}_k$ denote the set of shifts with strictly fractional values in $\mathbf{X}_k$, i.e. $0 < X_{S,k} < 1$. Let $\mathcal{B}_k$ denote the set of blocks which are not integrally assigned in $\mathbf{X}_k$. Note that $\mathcal{S}_k$ is precisely the set of shifts contained in the blocks $\mathcal{B}_k$.

Our high level idea is the following. As it is hard to guarantee that the constraint (2.8) will be maintained exactly for each time step upon rounding, we will instead enforce such a constraint *only for large time intervals* (as we will show, this can be ensured as the number of such constraints will be few). However, as the algorithm proceeds over iterations and fixes some shifts integrally to 0/1, a time slot (or) time interval may have very little (or no) broadcast among the fractional shifts in $\mathbf{X}_k$. To get around this, instead of directly working with time intervals, we will adopt a different approach and consider *intervals of transmissions made by the shifts in $\mathcal{S}_k$*. In fact, this step will be critical in ensuring that the Lovett-Meka result can be applied later. To this end, the following notation will serve useful.

(i) Let $\mathcal{N}_k$ denote the multi-set of all time slots where a transmission is made by a shift in $\mathcal{S}_k$, and let $N_k = |\mathcal{N}_k|$. That is, a time step t appears m_t times in $\mathcal{N}_k$ if m_t shifts in $\mathcal{S}_k$ transmit at t.
(ii) Let us order the transmissions in $\mathcal{N}_k$ in the increasing order of time (breaking ties arbitrarily). Let $\mathcal{O}^k_i$ denote the shift that makes the i^{th} transmission from $\mathcal{N}_k$.

(iii) For $1 \le j_1 \le j_2 \le N_k$, let $\mathcal{O}^k_{[j_1,j_2]}$ denote the *multiset* of shifts $\uplus_{i=j_1}^{j_2} \mathcal{O}^k_j$. For an interval of indices $J = [j_1, j_2]$, $\mathcal{O}^k_J$ is defined identically.

(iv) For a fractional solution $\mathbf{X}$ and a multiset C of shifts, let $\mathbf{X}(C) := \sum_{S \in C} X_S$ denote the total fractional broadcast made by the fractional solution with respect to the multiset C of shifts.

Replacing time intervals by J-intervals. Let us consider property (iii) of Theorem 2.1. As stated, property (iii) is required for a collection of $\Theta(T^2_{\max})$ constraints, for different choices of t_1 and t_2. This property can also be expressed as collection of constraints that bound $\mathbf{X}(C)$ for the multi-set of shifts $C = \mathcal{O}^k_J$ corresponding to intervals $J = [j_1, j_2]$, where j_1 is the small index of the transmission that is made at time t_1, and j_2 is the largest index of the transmissions made at time t_2. So, we will consider enforcing such a constraint for each J-interval $[j_1, j_2]$ for $j_1, j_2 \in [N_k]$. In what follows, we will approximately capture all these constraints by a family of $\Theta(N_k)$ constraints using the dyadic interval decomposition.

Grouping Transmissions into Dyadic Intervals. WLOG, let us assume that N_k is a power of 2, and say $N_k = 2^\ell$. Let $\mathcal{D}_k$ denote the family of dyadic intervals of $[N_k]$, i.e., all intervals of the form $(i2^\ell/2^j, (i+1)2^\ell/2^j]$ where $0 \le j \le \ell$, and $0 \le i \le 2^j - 1$. . Now, instead of rounding $\mathsf{LP}_{\mathsf{aux}}$ directly, we will work with the core LP $\mathsf{LP}_{\mathsf{core}}$ (defined below) which has time constraints only on the dyadic intervals, and later argue that a good solution for the core LP implies a good solution for $\mathsf{LP}_{\mathsf{aux}}$. Intuitively, this is because we can express any J-interval $[j_1, j_2]$ as the concatenation of $O(\log N_k)$ many dyadic intervals, and the core LP will ensure that the capacity constraints are satisfied on these intervals.

$$\sum_{S \in \mathcal{S}_k} C(S) y_S \le \sum_{S \in \mathcal{S}_k} C(S) X_{S,k} \quad (\mathsf{LP}_{\mathsf{core}}) \tag{2.10}$$

$$\sum_{S \in S(B)} y_S = 1 \qquad \forall B \in \mathcal{B}_k \tag{2.11}$$

$$\mathbf{y}(\mathcal{O}^k_J) \le \mathbf{X}_k(\mathcal{O}^k_J) \qquad \forall J \in \mathcal{D}^l_k \tag{2.12}$$

$$\mathbf{y}(\mathcal{O}^k_J) \le \mathbf{X}_k(\mathcal{O}^k_J) \qquad \forall J \in \mathcal{D}^s_k \tag{2.13}$$

$$y_S \ge 0 \qquad \forall S \in \mathcal{S}_k \tag{2.14}$$

In the LP and henceforth, $\mathcal{D}^s_k$ denotes the "small" dyadic intervals, i.e., $\mathcal{D}^s_k = \{J \in \mathcal{D}^k, |J| \le O(\log T_{\max})\}$, and $\mathcal{D}^l_k = \mathcal{D}_k \setminus \mathcal{D}^s_k$ denotes the "larger" dyadic intervals. Also, $\mathbf{y} = \{y_S\}$ are the variables, and $\mathbf{X}_k = \{X_{S,k}\}$ is the fractional solution we start with in the k^{th} iteration. While the constraints (2.12) and (2.13) are identical as written, our rounding algorithm will treat them differently.

Lovett-Meka Rounding of $\mathsf{LP}_{\mathsf{core}}$. We now run the Lovett-Meka rounding algorithm [27] (henceforth LM algorithm) on the above core LP. The Lovett-Meka algorithm was developed to find low discrepancy partial colorings for any set system, and matches the best non-constructive result due to the celebrated work of Spencer [32] and simplifies a previous constructive algorithm of Bansal [2]. However, these techniques are much more general and can be applied to a variety of problems besides the discrepancy problem. Recently, Rothvoss [31] used this to improve the approximation factor for Bin Packing to $O(\log \mathrm{OPT} \log\log \mathrm{OPT})$, beating a long-standing bound of Karmarkar and Karp [25]. In general, we can view the LM rounding as a more refined (albeit lossy) form of traditional iterative rounding, where we have no control over the structure of a basic feasible solution. Much like Rothvoss's rounding [31], our rounding heavily relies on the fine control which the LM rounding gives us, in moving to a partially-integral solution. We first present the LM algorithm's guarantees as a blackbox, and then apply it in our setting.

THEOREM 2.2. [2]*[Constructive partial coloring theorem [27]] Let $\mathbf{y} \in [0,1]^m$ be any starting point, $\delta > 0$ be an arbitrary error parameter, $v_1, \ldots, v_n \in \mathbf{R}^n$ vectors and $\lambda_1, \ldots, \lambda_n \ge 0$ parameters with*

$$\sum_{i=1}^n e^{-\lambda_i^2/16} \mathbf{1}_{\lambda_i > 0} \le \frac{m}{16}\,. \tag{2.15}$$

Also suppose at most $9m/16$ constraints have $\lambda_i = 0$. Then there is a randomized $\widetilde{O}((m+n)^3/\delta^2)$-time algorithm to compute a vector $\mathbf{z} \in [0,1]^m$ with

(i) $z_j \in [0, \delta] \cup [1-\delta, 1]$ for at least $m/32$ of the indices $j \in [m]$,

(ii) $|v_i \cdot \mathbf{z} - v_i \cdot \mathbf{y}| \le \lambda_i ||v_i||_2$, for each $i \in [n]$.

Firstly, by setting δ inverse-polynomially small, we can ignore it and assume that $1/32$ of the variables become integral in $\mathbf{z}$. Now, we apply this theorem to $\mathsf{LP}_{\mathsf{core}}$, with $\mathbf{y}$ as the starting point, the different constraints (2.10)-(2.13) corresponding to the vectors $v_1, \ldots, v_n$, and the following choice of λ parameters:

(i) Set λ value to 0 for the constraints (2.10), (2.11), and (2.12),

(iii) Set λ value to $\Omega(\sqrt{\log\log T_{\max}})$ for all smaller interval constraints (2.13).

[2]The original theorem does not allow $5m/8$ λ_i's to be 0, but we show how to adapt their proof to allow this setting in Appendix A.2

We now show that these parameters satisfy the condition (A.1) so that we can apply the LM rounding.

LEMMA 2.3. *The choice of λ above for the constraints (2.10)-(2.13) satisfies the conditions for Theorem 2.2.*

Proof. We first show that at most $9m/16$ constraints have $\lambda_i = 0$. Firstly, there is only one objective function constraint (2.10) which accounts for 1. Likewise, each block constraint (2.11) accounts for 1, and there are $|\mathcal{B}_k| \leq (1/2)|\mathcal{S}_k|$ of them. This is because each non-integral block has at least 2 shifts in $\mathcal{S}_k$. In total, they contribute at most $|\mathcal{S}_k|/2$.

Now, let us count the contribution of the large intervals. Now, since the dyadic intervals can be represented by a binary tree (root node has size N_k, and each child has half the size, and so on), the total number of large intervals (of size at least $\Omega(\log T_{\max})$) is at most $O(N_k/\log T_{\max})$. However, we note that $N_k = O(|\mathcal{S}_k| \log T_{\max})$. This is because each shift, say corresponding to block B and offset α and page p, makes at most $O(\log T_{\max})$ transmissions by Lemma 2.2, property (i). Therefore, the total contribution due to these intervals is at most $|\mathcal{S}_k|/32$, for a suitable choice of constants. In total, this is at most $9|\mathcal{S}_k|/16 = 9m/16$.

Next, we show that condition (A.1) is satisfied: for this, we focus on the small interval constraints. There are at most $O(T_{\max})$ such constraints, and each of these constraints has λ value $\Omega(\sqrt{\log\log T_{\max}})$. Therefore, the total contribution due to these constraints is $|\mathcal{S}_k|/16$, which is the RHS value of (A.1) as desired.

Therefore, by property (i) of Theorem 2.2, the LM rounding returns a solution $\mathbf{z}$ which has made at least $1/8^{th}$ of $\mathcal{S}_k$ integrally assigned. We define $\mathbf{X}_{k+1} := \mathbf{z}$, $\mathcal{S}_{k+1}$ denote the shifts which are still fractionally assigned in $\mathbf{X}_{k+1}$, $\mathcal{B}_{k+1}$ denote the blocks which are not integrally assigned, and repeat this process, until we end up with an integral solution. Let k^* denote the iteration when we end up with an integral solution.

We now analyze the final solution $\mathbf{X}_{k^*}$. The next lemmas prove properties (i) and (ii) of Theorem 2.1.

LEMMA 2.4. *The cost of the final solution $\sum_{S\in\mathcal{S}} C(S)X_{k^*,S} \leq \sum_{S\in\mathcal{S}} C(S)x^*_S$.*

Proof. Consider iteration $k \geq 0$. Notice that when we run the LM algorithm, we had set λ value to be 0 for constraint (2.10). Therefore, by property (ii) of Theorem 2.2, we have $\sum_{S\in\mathcal{S}_k} C(S)X_{k+1,S} - \sum_{S\in\mathcal{S}_k} C(S)X_{k_S} \leq 0$. Moreover, once a shift gets integrally assigned in $\mathbf{X}_{k+1}$, it is never altered in subsequent iterations. Therefore, we can inductively apply this property and complete the proof.

LEMMA 2.5. *At every iteration k, for every block $B \in \mathcal{B}$, we have $\sum_{S\in S(B)} X_{S,k} = 1$.*

Proof. This again follows because throughout our iterative rounding, we maintain the property $\sum_{S\in S(B)} X_{S,k} = X_{S,k+1}$ since we set λ value to be 0 for all the block constraints.

As a result, we pick exactly one shift/offset in each block in our final integral solution. We are left with bounding the backlog of $\mathbf{X}_{k^*}$.

LEMMA 2.6. *For any interval $I = [t_1, t_2)$ and any $0 \leq k < k^*$, $\sum_{S\in\mathcal{S}_k} X_{S,k+1}\mathbf{1}(S,I) \leq \sum_{S\in\mathcal{S}_k} X_{S,k} + O(B)$, where $B = \sqrt{\log T_{\max} \log\log T_{\max}}$.*

Proof. Consider some iteration $0 \leq k < k^*$ and time interval $I = [t_1, t_2]$. Firstly, notice that for any fractional solution $\mathbf{y}$, we can view the total fractional transmission by shifts in $\mathcal{S}_k$ in this time interval I (i.e., the set $\mathcal{N}_k \cap I$), as the dot product $\mathcal{O}^k_{J(I)} \cdot \mathbf{y}$, where $J(I)$ is an appropriately defined interval of the form $J(I) = [j_1, j_2]$, with $1 \leq j_1 \leq j_2 \leq N_k$. Notice that here, and for the rest of the proof, we are viewing the multiset $\mathcal{O}^k_J$ as a vector with integral entries in $|\mathcal{S}_k|$ dimensions.

In this view, we can rephrase the statement of the lemma as $\mathcal{O}^k_{J(I)} \cdot (\mathbf{X}_{k+1} - \mathbf{X}_k) \leq O(B)$. To this end, we decompose the interval $J(I)$ into a concatenation of intervals belonging to the dyadic family, and individually compute the error accrued by $\mathbf{X}_{k+1} - \mathbf{X}_k$ in each of these intervals. Moreover, the larger intervals J accrue no error, i.e., $\mathcal{O}^k_J \cdot (\mathbf{X}_{k+1} - \mathbf{X}_k) = 0$ because we set the λ parameter of all large intervals to 0.

Therefore we focus on the smaller intervals $J \in \mathcal{D}^s_k$. Since property (ii) of Theorem 2.2 says that the error $\mathcal{O}^k_J \cdot (\mathbf{X}_{k+1} - \mathbf{X}_k) \leq \lambda_J \cdot ||\mathcal{O}^k_J||_2$, we now focus on bounding the 2-norm of such constraints.

Indeed, since J is by definition a multiset consisting of $|J|$ contiguous transmissions, the ℓ_1 norm of $\mathcal{O}^k_J$ is $|J|$. Hence, in order to bound the ℓ_2 norm, it suffices to bound the ℓ_∞ norm. In general, however, the ℓ_∞ norm can be as large as $O(\log T_{\max})$, since the same shift can make multiple transmissions in an interval. While such an event can happen, we now appeal to the *interleaving nature of shifts* to reduce the ℓ_∞ norm. Indeed, by Lemma 6 (property (ii)), all shifts of a block make an equal number (upto ± 1) of transmissions in any time interval. Using this, we can actually eliminate many transmissions in this multiset, because our rounding preserves the sum $\sum_{S\in S(B)} X_{S,k}$ by Lemma 2.5. Therefore, we will be left with only at most 1 transmission per shift, thus bounding the ℓ_∞ norm by 1. More formally, we run the LM algorithm on the simplified family of constraints (2.13) as obtained below.

Simplifying Constraints (2.13). Consider some interval $J \in \mathcal{D}_k^s$, and some block $B \in \mathcal{B}_k$, and let $p = \min_{S \in S(B)} \mathcal{O}_J^k[S]$ denote the minimum number of transmissions done by shifts of block B in the multiset $\mathcal{O}_J^k$. Here, $\mathcal{O}_J^k[S]$ is the value of the coordinate corresponding to shift S in the multiset $\mathcal{O}_J^k$. Then, by Lemma 2.5, we know that $\sum_{S \in S(B)} y_S = 1$ is preserved by the final solution $\mathbf{X}_{k+1}$, and therefore we can subtract $p \sum_{S \in S(B)} y_S$ from the LHS of constraint (2.13), and p from the RHS before running the LM algorithm. Now, it is easy to see that if our rounded solution $\mathbf{X}_{k+1} = \mathbf{z}$ satisfies the simplified constraint with an additive error of Δ, then the original constraint is satisfied with the same error.

Claim 1. *The 2-norm of any constraint* (2.13) *corresponding to $J \in \mathcal{D}_k^s$ after simplification is $O(\sqrt{|J|})$.*

Proof. As mentioned above, the 1-norm of the original constraint is $O(|J|)$, and the simplification process only decreases the 1-norm. Moreover, the ∞-norm of the constraint is bounded by 2, since no shift can make more than 2 transmissions in the simplified constraint. The claim follows then by a simple norm inequality.

Therefore, the difference $\mathcal{O}_J^k \cdot (\mathbf{X}_{k+1} - \mathbf{X}_k) \leq \sqrt{|J| \log\log T_{\max}}$ for all small intervals $J \in \mathcal{D}_k^s$, since the λ value was set to $\Theta(\sqrt{\log\log T_{\max}})$ for such intervals. Since the small intervals form a geometric series of sizes $1, 2, 4, \ldots, \Theta(\log T_{\max})$, the total error of $\mathcal{O}_{J(I)}^k \cdot (\mathbf{X}_{k+1} - \mathbf{X}_k)$ is dominated by the largest interval, and hence is at most $O(\sqrt{\log T_{\max} \log\log T_{\max}})$, completing the proof.

To complete the proof of Theorem 2.1, we can apply Lemma 2.6 iteratively and bound the total backlog by $O(\log^{1.5} T_{\max} \sqrt{\log\log T_{\max}})$, thus proving property (iii). This completes the proof of Theorem 1.1.

3 Hardness of Approximating ℓ-permutations

Notation: Let $\chi : U \to \{\pm 1\}$ be a $\{\pm 1\}$ coloring of elements in U. For a subset $U' \subseteq U$, we define $\chi(U') = \sum_{u \in U'} \chi(u)$. For a sequence $p = (p_1, p_2, \cdots, p_m)$ of length m elements from U, define $\chi(p) = \sum_{i=1}^m \chi(p_i)$. Define $\mathsf{disc}_\chi(p) = \max_{p'} |\chi(p')|$, where p' ranges over all prefixes of p.

Let $\pi_1, \pi_2, \cdots, \pi_\ell$ be ℓ permutations of U. Then, the *discrepancy* and the *average discrepancy* of the permutation system $\Pi = \{\pi_1, \pi_2, \cdots, \pi_\ell\}$ are defined as

$$\mathsf{disc}(\Pi) := \min_{\chi : U \to \{\pm 1\}} \max_{i \in [\ell]} \mathsf{disc}_\chi(\pi_i),$$

$$\mathsf{avgdisc}(\Pi) := \min_{\chi : U \to \{\pm 1\}} \frac{1}{\ell} \sum_{i \in [\ell]} \mathsf{disc}_\chi(\pi_i).$$

In this section, we prove Theorem 1.5. Like the proof of Charikar et al. for discrepancy hardness for arbitrary set systems [9], our proof is also based on reduction from the 4-set splitting problem.

4-Set Splitting Problem. We are given a ground set U and a collection $\mathcal{S}$ of subsets of U where each $S \in \mathcal{S}$ has $|S| = 4$. The goal of the problem is to find a ± 1 coloring χ of U such that the number of sets $S \in \mathcal{S}$ with $\chi(S) = 0$ is maximized. We say a set $S \in \mathcal{S}$ is "equally split" by χ if $\chi(S) = 0$. So, we want to find a coloring χ so as to maximize the number of equally split sets. Additionally, we say that a 4-set splitting instance $(U, \mathcal{S})$ is Δ-restricted for some integer $\Delta > 0$, if each $u \in U$ appears in at most Δ sets in $\mathcal{S}$. W.l.o.g, we assume each $u \in U$ appears in at least 1 set in $\mathcal{S}$. Thus, we have $|U|/4 \leq |\mathcal{S}| \leq |U|\Delta/4$ in a Δ-restricted instance $(U, \mathcal{S})$. We start with the following hardness result (see, e.g., [9]).

Theorem 3.1. *There exist a real number $\delta \in (0,1)$ and an integer $\Delta \geq 1$ such that the following is true. Given a Δ-restricted 4-set splitting instance $(U, \mathcal{S})$, it is NP-hard to distinguish the following two cases:*

- **Yes Instance:** *There is a coloring χ such that all sets in $\mathcal{S}$ are equally split.*
- **No Instance:** *For every coloring χ, at most δ fraction of the sets in $\mathcal{S}$ are equally split.*

Since δ and Δ are absolute constants, we hide them in $O(.)$ and $\Omega(.)$ notations. Thus, we have $|\mathcal{S}| = \Theta(|U|)$ in a Δ-restricted instance $(U, \mathcal{S})$. The main lemma we shall prove in this section is the following. Combining this with Theorem 3.1 implies Theorem 1.5.

Lemma 3.1. *There exists some constants $\ell_0, C \in \mathbf{Z}_+$, such that: given a Δ-restricted 4-set splitting instance $(U, \mathcal{S})$ and an integer ℓ such that $\ell_0 \leq \ell \leq |U|/C$, there is an efficient randomized algorithm that constructs an ℓ-permutation instance $\Pi = \{\pi_1, \pi_2, \cdots, \pi_\ell\}$ with permutation size $O(\ell|U|)$, such that*

- *If $(U, \mathcal{S})$ is a yes instance, then $\mathsf{disc}(\Pi) = O(1)$ with probability 1.*
- *If $(U, \mathcal{S})$ is a no instance, then $\mathsf{avgdisc}(\Pi) = \Omega(\sqrt{\ell})$ with probability at least 1/2.*

We first sketch the proof of Lemma 3.1. Since the instance $(U, \mathcal{S})$ is Δ-restricted, we can partition $\mathcal{S}$ into 4Δ families of disjoint sets. For a fixed family $\mathcal{S}' \subseteq \mathcal{S}$, we produce $s = \Theta(\ell)$ permutations over $\bigcup_{S \in \mathcal{S}'} S$ randomly and independently. Each permutation $g = (g_1, g_2, \cdots, g_{4|\mathcal{S}'|})$ is obtained by concatenating the sets in $\mathcal{S}'$ according to a random permutation $(S_1, S_2, \cdots, S_{|\mathcal{S}'|})$ of $\mathcal{S}'$. That is, g satisfies $S_i = \{g_{4i-3}, g_{4i-2}, g_{4i-1}, g_{4i}\}$ for every $i \in [|\mathcal{S}'|]$. In total, we have constructed $4\Delta \times s = \Theta(\ell)$ "partial permutations" over U. If s is small enough, the number of partial permutations is at most ℓ.

For a no instance $(U, \mathcal{S})$ and any coloring χ of U, there is a family $\mathcal{S}'$ of size $\Omega(|\mathcal{S}|)$ in the partition such that a constant fraction of sets in $\mathcal{S}'$ are not equally split by χ. We split each of the s permutations for $\mathcal{S}'$ into blocks of length $4\ell'$, for some $\ell' = \Theta(\ell)$. We then expect that each block g' has $|\chi(g')| = \Omega(\sqrt{\ell})$ by "inverse Chernoff bound". Since each of the $s = \Theta(\ell)$ permutations contain $\Theta(|\mathcal{S}|/\ell')$ blocks, with probability at least $1 - e^{-\Omega(|\mathcal{S}|)}$, $s/2$ permutations contain bad blocks, i.e. blocks g' such that $|\chi(g)| \geq \Omega(\sqrt{\ell'})$. Thus, with probability at least $1 - e^{-\Omega(|\mathcal{S}|)}$, $s/2$ permutations g have $\mathsf{disc}_\chi(g) \geq \Omega(\sqrt{\ell'})$. Taking union bound over all colorings χ of U, we have that with probability at least $1/2$, the system of partial permutations has *average discrepancy* $\Omega(\sqrt{\ell})$. In the formal proof, we need to carefully consider the dependence between the blocks.

Notice that for a yes instance, the system of partial permutations has discrepancy 2, since every partial permutation is a concatenation of some sets in $\mathcal{S}$. Thus, we have almost proved Lemma 3.1 except that the permutations are partial. In the last and important step, we obtain a system of perfect permutations by appending elements to the end of partial permutations, so that the system for a yes instance still has $O(1)$ discrepancy.

The remaining part of this section is organized as follows. In Section 3.1, we construct the system of ℓ *partial permutations* for the given 4-set splitting instance $(U, \mathcal{S})$. Then in Section 3.2, we prove that the system for no instances has *average discrepancy* $\Omega(\ell^{1/2})$. Finally in Section 3.3, we show how to extend the partial permutations to full permutations so that the system for yes instances still has discrepancy $O(1)$.

3.1 Construction of partial permutations Since $(U, \mathcal{S})$ is a Δ-restricted instance, we can partition $\mathcal{S}$ into 4Δ families of disjoint sets. For each family $\mathcal{S}' \subseteq \mathcal{S}$ of disjoint sets, we construct $s := \lceil 64\Delta\ell'/(1-\delta)\rceil$ permutations over $\bigcup_{S \in \mathcal{S}'} S$, for some suitable parameter $\ell' = \Theta(\ell)$ (here δ is as defined in theorem 3.1). Each permutation is constructed randomly and independently as follows. Let $g = (g_1, g_2, \cdots, g_{4|\mathcal{S}'|})$ be the permutation over $\bigcup_{S \in \mathcal{S}'} S$, obtained by concatenating sets in $\mathcal{S}$ according to a random permutation $(S_1, S_2, \cdots, S_{|\mathcal{S}'|})$ over $\mathcal{S}'$. That is, $S_i = \{g_{4i-3}, g_{4i-2}, g_{4i-1}, g_{4i}\}$ for every $i \in [|\mathcal{S}'|]$. The 4 elements in S_i are arbitrarily ordered in g. Notice the sequence g is a *partial permutation* of U; that is, g does not need to contain all elements in U. In total, we have constructed $4\Delta \times s = O(\ell)$ partial permutations. We select ℓ' small enough so that the number of partial permutations is at most ℓ.

For a yes instance $(U, \mathcal{S})$ let $\chi : U \to \{\pm 1\}$ be the coloring that equally split all sets in $\mathcal{S}$. Then, it is easy to see that every partial permutation g has $\mathsf{disc}_\chi(g) \leq 2$, since g is a concatenation of sets in $\mathcal{S}$. Thus, for yes instances, the partial permutation system has maximum discrepancy $O(1)$.

3.2 $\Omega(\sqrt{\ell})$-discrepancy on the partial permutation system for no instances In this section, we show that if $(U, \mathcal{S})$ is a no instance, then with probability at least $1/2$, for every coloring $\chi : U \to \{\pm 1\}$, at least $\Omega(\ell)$ partial permutations have discrepancy at least $\Omega(\sqrt{\ell})$ w.r.t χ.

We start with the following technical lemma.

LEMMA 3.2. *Let $\mathcal{T} \subseteq \mathcal{S}$ be a subfamily of* disjoint *sets in $\mathcal{S}$ and $\chi : U \to \{\pm 1\}$ be a coloring of U such that at least $(1-\delta)|\mathcal{S}|/(8\Delta)$ sets in $\mathcal{T}$ are not equally split by χ. Let $S_1, S_2, \cdots, S_{\ell'}$ be ℓ'* different *sets selected randomly from $\mathcal{T}$. Then, with probability at least 0.8, we have $\left|\sum_{i \in [\ell']} \chi(S_i)\right| \geq \Omega(\sqrt{\ell})$.*

Proof. Let $X_i = \chi(S_i)$ for every $i \in [\ell']$. We first consider the simpler procedure where each S_i is selected from $\mathcal{T}$ randomly and independently (that is, we do not require the ℓ' sets to be different). Then $X_1, X_2, \cdots, X_{\ell'}$ are i.i.d. Let $\mu = \mathbb{E}[X_i]$ and w.l.o.g, assume $\mu \geq 0$.

Since $|\mathcal{T}| \leq |\mathcal{S}|$, at least $\frac{(1-\delta)}{8\Delta}$ fraction of sets in $\mathcal{T}$ are not equally split by χ. Then, with probability at least $(1-\delta)/(8\Delta)$, $|X_i| \geq 1$, we have $\mathbb{E}[X_i^2] \geq (1-\delta)/(8\Delta)$. Then, $\sigma^2 := \mathsf{V}ar(X_i) = \mathbb{E}[X_i^2] - \mu^2 \geq (1-\delta)/(8\Delta) - \mu^2 \geq (1-\delta)/(8\Delta) - 169/\ell'$. For large enough ℓ', $\sigma^2 \geq (1-\delta)/(10\Delta)$. We apply the following theorem to lower-bound the tail probability of $\chi(g)$.

THEOREM 3.2. (BERRY-ESSEEN INEQUALITY) *Let $X_1, X_2, \cdots, X_n$ be independent random variables with $\mathbb{E}(X_i) = 0, E(X_i^2) = \sigma_i^2 > 0$ and $\mathbb{E}(|X_i|^3) = \rho_i < \infty$. Let F be the cumulative density function of*

$$Y := \frac{\sum_{i=1}^n X_i}{\sqrt{\sum_{i=1}^n \sigma_i^2}}.$$

Then, there is an absolute constant C_1 such that

$$\sup_{x \in \mathbb{R}} |F(x) - \Phi(x)| \leq C_1 \left(\sum_{i=1}^n \sigma_i^2\right)^{-3/2} \cdot \sum_{i=1}^n \rho_i,$$

where Φ is the culmulative density function of the normal distribution.

We apply the theorem to the variables $X_i' := X_i - \mu$. Then, $\rho_i := \mathbb{E}(|X_i'^3|) \leq 8\mathbb{E}(X_i'^2) = 8\sigma^2$, since $|X_i'| \leq 8$

with probability 1. Then,

$$\begin{aligned}
&\Pr\left[\left|\sum_{i=1}^{\ell'} X_i\right| \geq 0.1\sigma\sqrt{\ell'}\right] \\
&= 1 - \Pr\left[-0.1 - \frac{\sqrt{\ell'}\mu}{\sigma} < \frac{\left|\sum_{i=1}^{\ell'} X_i'\right|}{\sigma\sqrt{\ell'}} < 0.1 - \frac{\sqrt{\ell'}\mu}{\sigma}\right] \\
&\geq 1 - \left(\Phi\left(0.1 - \sqrt{\ell'}\mu/\sigma\right) - \Phi\left(-0.1 - \sqrt{\ell'}\mu/\sigma\right)\right) \\
&\quad - 2C_1\left(\ell'\sigma^2\right)^{-3/2} \cdot \ell'\rho \\
&\geq 1 - (\Phi(0.1) - \Phi(-0.1)) - 16C_1/\sqrt{\ell'\sigma^2} \\
&\geq 0.92 - 32C_1/\sqrt{(1-\delta)\ell'}.
\end{aligned}$$

If ℓ_0 in Lemma 3.1 is large enough, the above probability is at least 0.9. Since $\sigma \geq \sqrt{(1-\delta)/(8\Delta)} = \Omega(1)$, we have $\Pr\left[\chi(g') \geq \Omega(\sqrt{\ell'})\right] \geq 0.9$.

The probability that $S_1, S_2, \cdots, S_{\ell'}$ are disjoint is at least

$$\frac{|\mathcal{T}|\left(|\mathcal{T}|-1\right)\left(|\mathcal{T}|-2\right)\cdots\left(|\mathcal{T}|-(\ell'-1)\right)}{|\mathcal{T}|^{\ell'}}.$$

If the constant C in Lemma 3.1 is large enough, we have $|\mathcal{T}| \geq 11\ell'$, the above quantity is at least $(1-1/11\ell')^{\ell'} \geq e^{-1/10} > 0.9$. Thus, in the sampling process without replacement, with probability at least $0.9+0.9-1 = 0.8$, the generated sequence g has $|\chi(g)| \geq \Omega(\sqrt{\ell'})$.

Now, focus on a no instance $(U, \mathcal{S})$ and an arbitrary coloring $\chi : U \to \{\pm 1\}$. Then, at least $(1-\delta)|\mathcal{S}|$ sets in $\mathcal{S}$ are not equally split by χ. Recall that we have partitioned $\mathcal{S}$ into 4Δ families of disjoint sets. At least $(1-\delta)|\mathcal{S}|/(4\Delta)$ sets in some family are not equally split by χ. Focus on such a family $\mathcal{S}'$ and let $(S_1, S_2, \cdots, S_{|\mathcal{S}'|})$ be a random permutation on $\mathcal{S}'$.

Split $(S_1, S_2, \cdots, S_{|\mathcal{S}'|})$ into blocks of sizes ℓ'. Consider a block $(S_{t\ell'+1}, S_{t\ell'+2}, \cdots, S_{t\ell'+\ell'})$. Notice that if $S_1, S_2, \cdots, S_{t\ell'}$ are fixed, then $S_{t\ell'+1}, S_{t\ell'+2}, \cdots, S_{t\ell'+\ell'}$ are ℓ' different sets, randomly selected from the subfamily $\mathcal{S}' \setminus \{S_1, S_2, \cdots, S_{t\ell'}\}$. If $t \leq (1-\delta)|\mathcal{S}|/(8\Delta\ell')$, then at least $(1-\delta)|\mathcal{S}|/(4\Delta) - (1-\delta)|\mathcal{S}|/(8\Delta) = (1-\delta)|\mathcal{S}|/(8\Delta)$ sets from the subfamily are not equally split by χ. By Lemma 3.2, with probability at least 0.8,

$$\left|\sum_{i=t\ell'+1}^{t\ell'+\ell'} \chi(S_i)\right| \geq \Omega(\sqrt{\ell}), \tag{3.16}$$

for any choices of $S_1, S_2, \cdots, S_{t\ell'}$. Thus, the probability that there exists a $t \leq (1-\delta)|\mathcal{S}|/(8\Delta\ell')$ satisfying (3.16) is at least $1 - 0.2^{(1-\delta)|\mathcal{S}|/(8\Delta\ell')}$. If we let g be the permutation over $\bigcup_{S\in\mathcal{S}'} S$ according to $(S_1, S_2, \cdots, S_{|\mathcal{S}'|})$, then $\mathsf{disc}_\chi(g) \geq \Omega(\sqrt{\ell})$ with probability at least $1 - 0.2^{(1-\delta)|\mathcal{S}|/(8\Delta\ell')}$.

Recall that in Section 3.1, we have constructed $s = \lceil 64\Delta\ell'/(1-\delta)\rceil$ permutations $g_1, g_2, \cdots, g_s$ over $\bigcup_{S\in\mathcal{S}'} S$ independently. With probability at most

$$\binom{s}{s/2} 0.2^{(1-\delta)|\mathcal{S}|/(8\Delta\ell')\times s/2} \leq \binom{s}{s/2} 0.2^{4|\mathcal{S}|} \leq 0.4^{|U|},$$

$s/2$ choices of $i \in [s]$ satisfy $\mathsf{disc}_\chi(g_i) \leq \Omega(\sqrt{\ell})$. The last inequality used the fact that $|\mathcal{S}| \geq |U|/4$ and $s \leq |U|$ (if the constant C in Lemma 3.1 is large enough).

By union bound over all colorings χ, with probability at least $1 - 2^{|U|} \cdot 0.4^{|U|} \geq 1/2$, for every coloring χ, at least $s/2 = \Omega(\ell)$ partial permutations have discrepancy $\Omega(\sqrt{\ell})$. In other words, with probability at least $1/2$, the system of the ℓ partial permutations has average discrepancy $\Omega(\sqrt{\ell})$.

3.3 Completing Partial Permutations We have almost proved Lemma 3.1, except that the obtained "permutations" are partial. The remaining task is to complete the partial permutations by appending elements to their ends. In the process, we introduce a new set V of elements. Then all the ℓ permutations will be on $U \cup V$. Notice that appending elements does not decrease the discrepancy. In particular, for a no instance $(U, \mathcal{S})$, the permutation system still has average discrepancy $\Omega(\sqrt{\ell})$. We need to ensure that in a yes instance, no matter what the coloring for U is, we can find a coloring for V so that the resulting permutation has $O(1)$ discrepancy.

To this end, we focus on a yes instance $(U, \mathcal{S})$ and a coloring χ that equally splits all sets in $\mathcal{S}$. We can assume $\chi(U) = 0$ by adding dummy elements to U. For each partial permutation, we append the elements of U that do not appear in the permutation arbitrarily to its end. We thus obtain ℓ permutations of U. Let $p_1, p_2, \cdots, p_\ell$ be the ℓ suffixes we appended to the end of the ℓ partial permutations. Since $\chi(U) = 0$, we have $\chi(p_i) = 0$ for every $i \in [\ell]$. Now, we insert the elements of V to each of the sequences p_i to form ℓ new sequences $q_1, q_2, \cdots, q_\ell$. We guarantee that no matter what the original coloring χ is, we can find a coloring χ' for V such that the discrepancy of the system $\{q_1, q_2, \cdots, q_\ell\}$ has $O(1)$ discrepancy w.r.t to χ and χ'. The following lemma is the crux of this step, which also completes the proof of Lemma 3.1.

Lemma 3.3. *There is a large enough constant d such that the following is true. Let U be the ground set and $p_1, p_2, \cdots, p_\ell$ be ℓ sequences of elements in U, each of even length. Then, we can efficiently construct a set V of $d \times \sum_{i=1}^{\ell} |p_i|$ elements with $U \cap V = \emptyset$ and ℓ sequences $q_1, q_2, \cdots, q_\ell$, such that* *(i) Each sequence q_i is formed by inserting all the elements of V into p_i;*

(ii) For every coloring $\chi : U \to \{\pm 1\}$ s.t $\chi(p_i) = 0$ for every $i \in [\ell]$, we can find a coloring $\chi' : U \cup V \to \{\pm 1\}$ such that $\chi'|_U \equiv \chi$ and $\mathsf{disc}_{\chi'}(q_i) \le \frac{d+1}{2}$ for every $i \in [\ell]$.

Proof. In our construction, we shall apply some explicit construction of expanders. Recall that the expansion of a graph $G = (V, E)$ is

$$\Phi(G) = \min_{S \subseteq V; |S| \le |V|/2} \frac{|E(S, V \setminus S)|}{|S|} .$$

Let d be a large enough odd number such that given any even number $n \ge 0$, we can construct in $\mathsf{poly}(n)$ time a d-regular graph with n vertices and expansion at least 1. It is known that such constructions exist.

We describe how we obtain the sequences $q_1, q_2, \cdots, q_\ell$ from $p_1, p_2, \cdots, p_\ell$. Let $m_i = |p_i|$, and $p_{i,j}$ denote the j^{th} element of the sequence p_i. In the first step, for each $i \in [\ell]$ and $j \in [m_i]$, we insert d unique elements right before $p_{i,j}$ in the sequence p_i. Thus, we have inserted $|V| = d \sum_{i=1}^{\ell} m_i$ elements across all sequences, and also guarantee that each element is inserted exactly once. We use $A_{i,j}$ to denote the set of d elements inserted before $p_{i,j}$. Then, the set $\{A_{i,j} : i \in [\ell], j \in [m_i]\}$ forms a partitioning of V into groups of size d. Let $\mathcal{A}_i = \bigcup_{j \in [m_i]} A_{i,j}$.

In order to satisfy Property (i), we need to insert the elements inserted into p_i in the other sequences also. We now do this in a careful manner. To this end, for each $i \in [\ell]$, we divide the dm_i elements of $\mathcal{A}_i$ into $dm_i/2$ pairs, in the manner described in the next paragraph (recall that m_i is even). Let $\mathcal{B}_i$ denote the set of pairs. For every $i' \in [\ell], i' \ne i$, we append the dm_i elements to the end of $p_{i'}$, with the only constraint that the 2 elements of each pair in $\mathcal{B}_i$ appear next to each other in the appended sequence.

We now describe the requirement that our pairing $\mathcal{B}_i$ must satisfy. To this end, we construct a d-regular graph $H_i = ([m_i], E_{H_i})$ as follows. Each vertex $j \in [m_i]$ corresponds to the set $A_{i,j}$. For each pair $\{u, v\} \in \mathcal{B}_i$ with $u \in A_{i,j}$ and $v \in A_{i,j'}$, there is an edge (j, j') in E_{H_i}. The only requirement for $\mathcal{B}_i$ is that the associated d-regular graph H_i has expansion at least 1. Moreover, we can easily recover a pairing from any d-regular graph on m_i vertices with expansion at least 1. This finishes the construction of the sequences $q_1, \cdots, q_\ell$.

Now, we show how we extend a coloring $\chi : U \to \{\pm 1\}$ to a coloring for $\chi' : U \cup V \to \{\pm 1\}$. Suppose we are given a coloring $\chi : U \to \{\pm 1\}$ such that $\chi(p_i) = 0$ for every $i \in [\ell]$. Consider the element $p_{i,j}$ and the d elements $A_{i,j}$ inserted before $p_{i,j}$. We require that the $d+1$ elements are equally split by χ'. Thus, we require $\chi'(A_{i,j}) = -\chi(p_{i,j})$. Then, for each pair $\{u, v\} \in \mathcal{B}_i$, we require $\chi'(u) + \chi'(v) = 0$. The first condition will ensure that the intervals we add before each $p_{i,j}$ have no discrepancy, and the second condition ensures that the pairs we add at the end of each suffix also don't have any discrepancy. Therefore, these two requirements are sufficient to guarantee that the sequences $\{q_i\}$ have $O(1)$-discrepancy.

To this end, notice that the requirements for different i's are independent and thus we can focus on a fixed $i \in [\ell]$. Then, the requirements for χ' and the set $\mathcal{A}_i$ are:

$$\sum_{u \in A_{i,j}} \chi'(u) = -\chi(p_{i,j}) \qquad \forall j \in [m_i]$$
$$\chi'(u) + \chi'(v) = 0 \qquad \forall B = \{u, v\} \in \mathcal{B}_i$$

For an element $u \in \mathcal{A}_i$, let $y_u = \frac{\chi'(u)+1}{2}$. We can then convert the above linear system to a system on y variables:

$$\sum_{u \in A_{i,j}} y_u = \frac{d - \chi(p_{i,j})}{2} \qquad \forall j \in [m_i]$$
$$y_u + y_v = 1 \qquad \forall \{u, v\} \in \mathcal{B}_i$$

We prove that the above linear system has a $\{0, 1\}$ solution. To this end, focus on the d-regular graph H_i. To gain some intuition, let us see what the above LP is doing. Indeed, it tries to orient the edges of H so that the in-degree of j is exactly $\frac{d-\chi(p_{i,j})}{2}$ for every $j \in [m_i]$. To see this, for each edge corresponding to $\{u, v\} \in \mathcal{B}_i, u \in A_{i,j}, v \in A_{i,j'}$, we let $y_u = 1, y_v = 0$ if the edge is oriented from j' to j and let $y_u = 0, y_v = 1$ otherwise.

Now, because $\chi(p_i) = 0$, we have that the number of edges in H_i is $dm_i/2 = \sum_{j=1}^{m_i} \frac{d-\chi(p_{i,j})}{2}$. Therefore, it is an equivalent problem to require that the in-degree of j is *at most* $\frac{d-\chi(p_{i,j})}{2}$ for every $j \in [m_i]$. By viewing this as a bipartite matching problem, it is easy to see that there exists a valid orientation if and only if for every subset $S \subseteq [m_i]$, $|E_{H_i}(S)| \le \sum_{j \in S} \frac{d-\chi(p_{i,j})}{2}$. Here, $E(S)$ is the set of edges induced inside the subset S of vertices.

Indeed, if $|S| \le m_i/2$, by the expansion property of H_i, we have $|E_{H_i}(S)| \le \frac{d|S|-|S|}{2} = \sum_{j \in S} \frac{d-1}{2} \le \sum_{j \in S} \frac{d-\chi(p_{i,j})}{2}$. On the other hand, if $|S| \ge m_i/2$, we have $|E_{H_i}(S)| \le \frac{d|S|-(m_i-|S|)}{2} = \frac{(d+1)|S|-m_i}{2}$. But notice that $\sum_{j \in S} \frac{d-\chi(p_{i,j})}{2} \ge \frac{m_i}{2} \times \frac{d-1}{2} + \left(|S| - \frac{m_i}{2}\right) \times \frac{d+1}{2} = \frac{(d+1)S-m_i}{2}$, since there are at most $m_i/2$ $p_{i,j}$'s with color 1. Thus, we have $|E_{H_i}(S)| \le \sum_{j \in S} \frac{d-\chi(p_{i,j})}{2}$

in both cases. Therefore, the above system has a $\{0,1\}$ solution, which implies that the linear system with χ' variables has a $\{\pm 1\}$ solution.

4 Integrality Gaps and Hardness of Broadcast Scheduling

In this section, we show hardness of approximation for the broadcast scheduling problem, via a reduction from the discrepancy of ℓ-permutations problem. Indeed, given an instance $\mathcal{I}_{\mathsf{perm}}$ of ℓ permutations $\pi'_1, \pi'_2, \pi'_3, \ldots, \pi'_\ell$ over $[m']$, our broadcast scheduling instance $\mathcal{I}_{\mathsf{bcast}}$ is then constructed as follows.

Construction of Instance $\mathcal{I}_{\mathsf{bcast}}$. For a suitable choice of parameter M to be determined later, we first concatenate M unique copies (on different sets of variables) of each permutation π'_i to generate permutation π_i over $[2m'M]$. In what follows, let $m := 2m'M$, an even integer, be the size of the permutations π_i. The time horizon in $\mathcal{I}_{\mathsf{bcast}}$ is divided into $2\ell+1$ permutation intervals (P-interval for short) $P_1, P_2, \cdots, P_{2\ell+1}$ and $2\ell+1$ *forbidden intervals* (F-interval) $F_1, F_2, \cdots, F_{2\ell+1}$. The P-intervals and F-intervals are alternately placed in the time horizon in the form $P_1, F_1, P_2, F_2, \ldots, F_{2\ell+1}$. Let $P_{i,j}$ and $F_{i,j}$ denote the j-th time slot in P_i and F_i respectively. The construction will often involve a parameter D which we set to be $m/2$ with the knowledge of hindsight.

Defining P-intervals. Each P-interval has length $m/2$. For $i \in [2\ell], j \in [m/2]$, we request 2 pages $\pi_{\lceil i/2\rceil, 2j-1}$ and $\pi_{\lceil i/2\rceil, 2j}$ of weight $1/D^{i-1}$ in time slot $P_{i,j}$. There are no requests in P-interval $P_{2\ell+1}$.

Defining F-intervals. For $i \in [2\ell+1]$, the F-interval F_i has length mD^{i-1}. At each time slot in an F-interval, we request a page which is requested only once in the instance, of weight m/D^{i-1}.

This completes the construction, see Figure 1. We now start with some useful claims.

CLAIM 2. *The total weight of requests in F-intervals is $\Theta(\ell m^2)$, and that of requests in P-intervals is $\Theta(m)$.*

Proof. The total weight of requests in F-intervals is $\sum_{i=1}^{2\ell+1}\left(mD^{i-1} \times m/D^{i-1}\right) = \sum_{i=1}^{2\ell+1} m^2 = (2\ell+1)m^2 = \Theta(\ell m^2)$. The total weight of requests in P-intervals is $\sum_{i=1}^{2\ell} m(1/D^{i-1}) = \Theta(m)$.

CLAIM 3. *There is a fractional scheduling whose average response time is $\Theta(1)$.*

Proof. In a time slot a F-interval, we broadcast the F-page requested in the slot. In a time slot of P_i, $i \in [2\ell]$, we broadcast $1/2$ fraction of each P-page requested in the slot. In $P_{2\ell+1}$, we broadcast $1/2$ fraction of all pages in $[m]$ in an arbitrary way. Then, each request in a F-interval has response time 1. Consider a request in P_i. $1/2$ fraction of the request is satisfied immediately, with response time 1, and the other $1/2$ fraction is satisfied at some lot in P_{i+1}, with response time $\Theta(mD^{i-1})$. Thus, the total weighted response time is $\Theta(\ell m^2) \times 1 + \sum_{i=1}^{2\ell}\left(2m \times 1/D^{i-1} \times \Theta(mD^{i-1})\right) = \Theta(\ell m^2)$, and the average response time is $\Theta(1)$.

4.1 Completeness

In this section, we show the following theorem.

THEOREM 4.1. *If the system of ℓ-permutations has average discrepancy C, then there is a broadcast schedule for which the average response time is $O(C)$.*

Consider a balanced coloring χ of the permutation system, i.e., there are an equal number of $+1$'s and -1's. This is WLOG, as the imbalance is at most C, and hence changes the overall discrepancy by at most a factor of 2. For $i \in [\ell]$, let c_i denote the discrepancy of the i^{th} permutation w.r.t χ. Then $C = \sum_{i\in[\ell]} c_i/\ell$.

We now derive the broadcast schedule in a natural manner: Firstly, we simply broadcast each page in the F-intervals as and when the requests are released. Then, our idea is to schedule the $m/2$ elements labeled $+1$ in the odd P-intervals, and those labeled -1 in the even P-intervals. Indeed, for odd P-intervals, say P_{2i-1}, we schedule the elements of $[m]$ labeled $+1$ *in the order in which they appear in* π_i. Likewise, we schedule the elements with -1 label in even P-intervals P_{2i} in the order of π_i. Moreover, within each P_i, we *right-shift* the schedule by $c_{\lceil i/2\rceil}$ time steps in order to account for the discrepancy, with a wrap-around to the beginning of P_i. E.g., if the original schedule is $2, 7, 10, 9, 5$ and the discrepancy of the permutation was 2, the new schedule would be $9, 5, 2, 7, 10$. This gives us the final schedule.

Let $\chi_+ = \{i \in [m] : \chi(i) = 1\}$ and $\chi_- = [m] \setminus \chi_+$. The following lemma shows that most requests in P_i are either satisfied in P_i or P_{i+1}. This crucially uses the fact that the discrepancy of $\pi_{\lceil i/2\rceil}$ is $c_{\lceil i/2\rceil}$. The following lemmas complete the proof of Theorem 4.1.

LEMMA 4.1. *For any odd P-interval P_{2i-1}, the requests corresponding to elements in χ_- are satisfied in P_{2i}, and all but the last c_i requests in χ_+ corresponding to π_i are satisfied in P_{2i-1} itself. An analogous statement holds for even P-intervals with the signs flipped.*

Proof. The first part follows trivially, because all the elements in χ_- are broadcast in P_{2i}. Now let us focus on the requests in χ_+, and suppose there exists a request (which is not in the last c_i requests) that is not satisfied in P_{2i-1}. Let us consider the last such request r, say corresponding to element $\pi_{i,j}$. The fact that r is

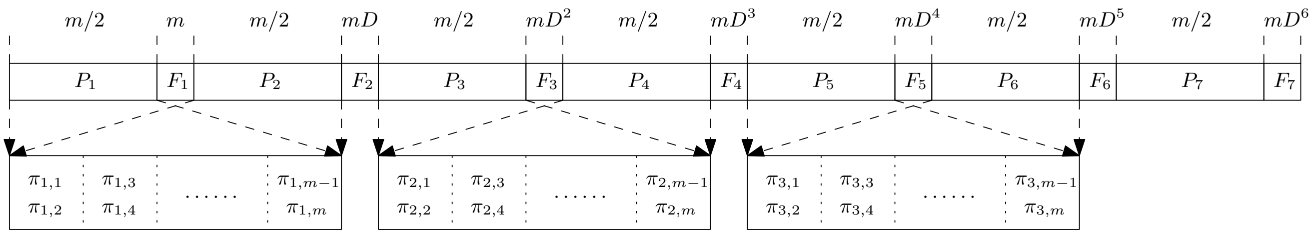

Figure 1: Constructing the broadcast scheduling instance from the ℓ-permutations instance

not satisfied in P_{2i-1} means that there are at least $\lceil (m-j)/2 \rceil + c_i$ pages in χ_+ scheduled among the elements $\pi_{i,j}, \pi_{i,j+1}, \ldots, \pi_{i,m}$. Otherwise, since there are $\lceil (m-j)/2 \rceil$ slots in P_{2i-1} after the arrival time of the request r, and we *right-shift* the schedule by c_i units, r would be satisfied within P_{2i-1} itself.

But this gives us a witness to the fact that the suffix $\pi_{i,j}, \pi_{i,j+1} \ldots, \pi_{i,m}$ has discrepancy at least $2c_i$, since there are at least $\lceil (m-j)/2 \rceil + c_i$ from χ_+ from these elements, and hence there can be at most $\lceil (m-j)/2 \rceil - c_i$ from χ_-. This contradicts the fact that the discrepancy of π_i under χ is c_i.

Lemma 4.2. *The total weighted response time of the above solution is $O(C\ell m^2)$.*

Proof. Each request in F-interval has a response time of 1, and hence their total weighted response time is $\Theta(m^2 \ell)$. Now let us consider the requests in some P-interval P_{2i-1} (an analogous argument works for the requests in P_{2i}). Each request corresponding to elements in χ_- are only satisfied in P_{2i} and incurs a response time of $\Theta(mD^{i-1})$. Therefore the total weighted response time of requests in χ_- is $m/2 \times 1/D^{i-1} \times \Theta(mD^{i-1}) = \Theta(m^2)$ (there are $m/2$ requests, each of weight $1/D^{i-1}$). Now let us focus on the requests in χ_+: all but c_i of them are satisfied in P_{2i-1} itself, giving a total weighted response time of $m/2 \times 1/D^{i-1} \times \Theta(m) = o(m^2)$. The c_i unsatisfied requests are satisfied the next time the pages in χ_+ are broadcast in P_{2i+1}. Their weighted response time is bounded by $c_i \times 1/D^{i-1} \times \Theta(mD^i) = \Theta(c_i m D) = \Theta(c_i m^2)$.

Therefore, the total weighted response time of requests in P_{2i-1} (and similarly, P_{2i}) is at most $O(c_i m^2)$. Summing over all i completes the proof.

4.2 Soundness In this section, we show the following theorem.

Theorem 4.2. *Given any broadcast schedule τ of average response time at most C, we can recover a coloring for the ℓ-permutations instance with average discrepancy $O(C)$.*

The rest of this section is devoted to the proof of the above theorem. Our goal is to show that most of the pages are in fact scheduled in alternating P-intervals, from which we can recover a coloring for the ℓ-permutation instance. To this end, we consider a *relaxed objective function.*

Definition 1. *In our new objective function, if a request in P_i is not satisfied by the end of F_{i+1} by a schedule τ, then it is automatically satisfied at that time, incurring a response time until the end of F_{i+1}, and no broadcast is needed to satisfy the request.*

Note that the average response time of a schedule can only decrease with this modified objective function (and hence is at most C). The following lemma says that we can assume in each slot of an F-interval, we broadcast the page requested.

Lemma 4.3. *Given any schedule τ for the instance, we can construct a schedule τ' such that*

- *The (relaxed) cost of τ' is at most the cost of τ.*
- *In each time slot of an F-interval, τ' broadcasts the corresponding F-page requested.*

Proof. Consider the first slot $F_{i,j}$ in an F-interval when τ does not broadcast the requested page f requested. Let us focus on all the unsatisfied requests at the beginning of the slot, and define the weight of a page to be the total weight of all unsatisfied requests for the page at this time.

We first claim that page f has the largest weight. Indeed, due to our assumption on $F_{i,j}$ being the first slot when τ does not broadcast the corresponding page, all previous requests in F-intervals are already satisfied. Thus suffices to compare f with pages in $[m]$. Now, because we relaxed our objective function, each page in $[m]$ has at most *two unsatisfied requests*: one from P_i and the other from P_{i-1}. Thus the weight of any page is at most $1/D^{i-2} + 1/D^{i-1} \leq 2/D^{i-2} = m/D^{i-1}$ (since $D = m/2$), which is the weight of f.

Now, suppose f is satisfied at some time $t > F_{i,j}$. Then we can simply switch the two broadcasts at time

$F_{i,j}$ and t. This does not increase the overall objective function, because (i) f is never requested after $F_{i,j}$, and the overall weighted response time only decreases. This completes the proof.

Therefore, we may WLOG assume that each request in an F-interval is satisfied immediately by our schedule τ'. We also assume each page is broadcast in each P-interval at most once: otherwise, we only keep the last broadcast and this increases the average response time by $o(1)$. Now, the relaxed objective implies that any request in P_i is either (i) satisfied in P_i or P_{i+1}, or (ii) automatically declared as satisfied at the end of F_{i+1} according to our relaxed objective — such requests are denoted as *bad requests*.

Claim 4. *The number of bad requests is at most $O(C\ell)$.*

Proof. A bad request (say from P_i) incurs a weighted response cost of at least $1/D^{i-1} \times \Theta(mD^i) = \Theta(m^2)$. The claim follows since the total weighted response time of the solution is $O(C\ell m^2)$.

Following on the notion of bad requests, we now categorize pages as either good or bad.

Definition 2. *A page is said to be "good" if it satisfies the following properties:*

(i) It is broadcast in alternate P-intervals, i.e., either the odd intervals or even intervals.
(ii) For all i, if the page is broadcast in P_i, it is broadcast after the corresponding request for the page arrived in P_i, which depends on $\pi_{\lceil i/2 \rceil}$.

All other pages are said to be "bad".

We will now show that most of the pages have to be scheduled in alternating P-intervals, based on the following intuition: (i) broadcasting the same page in consecutive P-intervals takes up too much space, as two consecutive P-intervals have a collective space of only m, one slot per element, and (ii) not making a broadcast in two consecutive P-intervals will create a bad request.

Lemma 4.4. *Given any schedule τ', we can construct a schedule τ'' such that: (i) no page is broadcast in consecutive P-intervals, (ii) the relaxed cost of τ'' is $O(C)$, and (iii) the total number of bad pages is $O(\ell C)$.*

Proof. We begin by making the following definitions and observations: Let $X_{i,j}$, for $1 \le i \le 2\ell + 1$ and $1 \le j \le m/2$ denote the number of requests which the broadcast τ' satisfies at time $P_{i,j}$. Notice that any broadcast in P_1 can satisfy at most 1 request. Likewise, a broadcast in $P_{2\ell+1}$ can satisfy at most one request, the one which arrived in $P_{2\ell}$ (requests in $P_{2\ell-1}$ can't be satisfied due to our relaxed objective function in Definition 1). Any other broadcast, say in P_i, can satisfy at most 2 requests, the ones on P_i and P_{i-1} for the corresponding page (earlier requests are again disallowed by Definition 1). We can therefore assign the trivial bounds $B_{i,j}$ on $X_{i,j}$ where $B_{i,j} = 1$ if $i = 1$ or $i = 2\ell + 1$ and $B_{i,j} = 2$ otherwise.

Using the above bounds, observe that the total number of requests that can be satisfied is at most

$$\sum_{i,j} B_{i,j} = (2\ell - 1)m + m = 2\ell m \tag{4.17}$$

On the other hand, there are $2\ell m$ requests of which $O(C\ell)$ are bad by Claim 4 and need not be served. Therefore,

$$\sum_{i,j} X_{i,j} \ge 2\ell m - O(C\ell) \tag{4.18}$$

$$\Rightarrow \sum_{i,j} (B_{i,j} - X_{i,j}) \le O(C\ell). \tag{4.19}$$

We first argue that we don't have too many consecutive P-intervals in which the same page is broadcast, and moreover, we show that we can also avoid such consecutive broadcasts.

Bounding Consecutive Broadcasts. Suppose a page p is scheduled in consecutive P-intervals $P_i, P_{i+1}, \ldots, P_{i+q}$. Let $j(p, i')$ denote the timeslot within $P_{i'}$ where p is broadcast. Then the total number of requests these broadcasts can serve is $\sum_{i'=i}^{i+q} X_{i,j(p,i')} \le q + 2$, since they can only serve requests which arrive in $P_{i-1}, P_i, \ldots, P_{i+q}$. However, the sum $\sum_{i'=i}^{i+q} B_{i,j(p,i')} = 2(q + 1)$ in (4.17). A boundary case is when the interval starts or ends at P_1 or $P_{2\ell+1}$, in which case both sums drop by one. Thus each consecutive interval contributes its length towards (4.19), which means that the sum of lengths of all such intervals is $O(C\ell)$, by inequality (4.19).

We now delete all but the last broadcast in every interval of consecutive broadcasts: for an interval of $q+1$ consecutive broadcasts, this will create q bad requests (each of which incurs $\Theta(m^2)$ weighted response time in the relaxed objective). Since their total sum of lengths is bounded by $O(C\ell)$, the total weighted response time increases by $O(C\ell m^2)$, and the average response time only increases by $O(C)$.

This proves (i) and (ii) of the claim.

Bounding Skipped Intervals. We now bound the number of bad pages. To this end, note that any time a page is not broadcast in two consecutive P-intervals P_i and P_{i+1}, there is a bad request which arrived in P_i.

Hence the number of such skipped broadcasts is at most $O(C\ell)$. Every other page make broadcasts in alternate P-intervals.

Bounding Bad Pages. Consider the $m - O(C\ell)$ pages which make broadcasts in alternate P-intervals. Of these pages, if a page makes a broadcast in an interval P_i before the corresponding request is made (according to the permutation $\pi_{\lceil i/2 \rceil}$), then it satisfies one fewer request that its bound $B_{i,j}$, and contributes one to (4.19). Therefore, there can be at most $O(C\ell)$ such pages. By definition, every other page is good, and this completes the proof.

Now we are ready to obtain a coloring of most of the elements of $[m]$, from which we will decode a coloring for one of the M copies of the ℓ-permutations system.

Decoding a Coloring. Consider all the good pages. If a good page is broadcast in the odd P-intervals, color it $+1$; if it is broadcast in even P-intervals, label it -1. We now use the fact that each permutation was in fact a concatenation of M copies of the original permutation to recover a copy without any bad pages. Indeed, if $M = \Omega(C\ell)$ is large enough, then at least one copy, say the q^{th} copy, has no bad pages. We will now show that our coloring for this copy has average discrepancy at most $O(C)$ w.r.t the original permutations.

Claim 5. *If permutation π'_i has discrepancy c_i according to our coloring of the q^{th} copy as defined above, then at least $\Omega(c_i)$ bad requests arrive in P-intervals $P_{2i-1} \cup P_{2i}$.*

Proof. Let us focus on the permutation copies $q, q+1, \ldots, M$. By the way we have constructed the broadcast instance, their requests appear contiguously in a suffix of the P-intervals P_{2i-1} and P_{2i}. Moreover, suppose some suffix of length t of the original permutation π'_i has discrepancy c_i w.r.t our coloring of the q^{th} copy. WLOG, let the number of $+1$'s in this suffix w.r.t our coloring be at least $(t + c_i)/2$.

Then consider the last $t/2 + (M - q)m'$ time slots of P_{2i-1}. Note that all the pages colored $+1$ in the q^{th} copy are requested in these time slots. Therefore, since all pages in the q^{th} copy are good, at least $(t + c_i)/2$ broadcasts for pages in q^{th} copy are in these slots. Therefore, at least $c_i/2$ slots are taken from the last $(M - q)m'/2$ slots of P_{2i-1}. Then consider the last $(M-q)m'/2$ slots in P_{2i-1} and the last $(M-q)m'/2$ slots in P_{2i}. At least $c_i/2$ pages out of the $(M - q)m'$ pages corresponding to copies $q + 1, \ldots, M$ are not broadcast in these slots. For each of these pages, either it is not broadcast in $P_{2i-1} \cup P_{2i}$ or it is before its corresponding request is released. Both cases lead to a bad request, and this completes the proof.

Since there are at most $O(C\ell)$ bad requests, we obtained a coloring with average discrepancy $O(C\ell/\ell) = O(C)$. This finishes the proof of Theorem 4.2.

Finally, we need to set the correct parameters to finish the proof of Theorems 1.3 and 1.4.

We now finish the proofs of Theorem 1.3 and 1.4. For Theorem 1.3, we take the 3-permutation instance (π_1, π_2, π_3) of permutation size n and discrepancy $\Omega(\log n)$ in [28]. Then, we apply our construction to obtain a broadcast scheduling instance with number of time slots being $T = O(n^7)$ and number of pages being at most $N = \Theta(T)$. By Claim 3, the LP solution for the instance has cost $\Theta(1)$. By Theorem 4.2 and the fact that the 3-permutation system has average discrepancy $\Omega(\log n)$, any integral scheduling of the instance has cost $\Omega(\log n) = \Omega(\log N)$. This finishes the proof of Theorem 1.3.

For Theorem 1.4, we start from a Δ-uniform 4-set splitting instance $(U, \mathcal{S})$ of size $n = |U|$. We let $\ell = \log^t n$ for some constant t. We construct ℓ permutations $\pi_1, \pi_2, \cdots, \pi_\ell$ of length $O(\ell n)$ by applying Lemma 3.1. Using the construction in the section, we obtain a broadcast scheduling instance of size $N = \Theta\left((n\ell)^{2\ell+1}\right)$. Thus, $\log N = (2\ell + 1)\log(\ell n) = \Theta(\log^{t+1} n)$. If $(U, \mathcal{S})$ is a yes instance, the permutation system $\Pi = (\pi_1, \pi_2, \cdots, \pi_\ell)$ has $\mathsf{disc}(\Pi) = O(1)$ and thus there is a scheduling of average response time $O(1)$ for the broadcast scheduling instance. If $(U, \mathcal{S})$ is a no instance, we have $\mathsf{avgdisc}(\Pi) = \Omega(\ell^{1/2})$. Thus, any scheduling for the broadcast scheduling instance has average response time $\Omega(\ell^{1/2})$. If there is a $N^{O(1)}$-time algorithm that approximates the cost of the broadcast scheduling instance within a factor of $O\left(\ell^{1/2}\right) = O\left(\log^{t/2} n\right) = O\left(\log^{t/2(t+1)} N\right)$, then there is a randomized $O\left(N^{O(1)}\right) = 2^{O(\log^{t+1} n)}$-time algorithm that distinguish between yes instances and no instances for Δ-uniform 4-set splitting. Assuming NP is not contained in $\mathsf{BPTIME}\left(2^{\log^t n}\right)$ for any constant t, there is no $O\left(\log^{1/2-\epsilon} n\right)$ approximation algorithm for broadcast scheduling instances of size n, for any $\epsilon > 0$. This finishes the proof of Theorem 1.4.

5 Acknowledgements

We thank an anonymous reviewer for suggesting a modification to our hardness construction that improved the hardness result from $\Omega(\log^{1/4-\epsilon} n)$ to $\Omega(\log^{1/2-\epsilon} n)$.

References

[1] Airmedia website, http://www.airmedia.com.

[2] N. Bansal. *Constructive Algorithms for Discrepancy Minimization.* Foundations of Computer Science, FOCS 2010, 3–10.

[3] N. Bansal, M. Charikar, S. Khanna and J. Naor. *Approximating the average response time in broadcast scheduling.* Proc. of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005.

[4] N. Bansal, D. Coppersmith and M. Sviridenko. *Improved approximation algorithms for broadcast scheduling.* SIAM Journal on Computing, 38(3), pp. 1157-1174, 2008.

[5] N. Bansal, R. Krishnaswamy, and V. Nagarajan. *Better scalable algorithms for broadcast scheduling.* In Automata, Languages and Programming (ICALP), pages 324-335, 2010.

[6] A. Bar-noy, S. Guha, Y. Katz, J. Naor, B. Schieber and H. Shachnai, Throughput Maximization of Real-Time Scheduling with Batching, In Proc. of Soda 2002, pp.742-751.

[7] Y. Bartal and S. Muthukrishnan. *Minimizing maximum response time in scheduling broadcasts.* Proc. of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 558-559, 2000.

[8] A. Blum, P. Chalasani, D. Coppersmith, W. R. Pulleyblank, P. Raghavan, and M. Sudan. *The minimum latency problem.* In Proc. 26th Symp. Theory of Computing (STOC), pages 163–171, 1994.

[9] M. Charikar, A. Newman and A. Nikolov. *Tight hardness for minimizing discrepancy.* In Proc. 22nd Symposium on Discrete Algorithms (SODA), pages 1607-1614, 2011.

[10] J. Chang, T. Erlebach, R. Gailis and S. Khuller. *Broadcast Scheduling: Algorithms and Complexity.* Proc. 19th ACM-SIAM Symp. on Disc. Algorithms, pages 473–482, 2008.

[11] M. Charikar and S. Khuller, A Robust Maximum Completion Time Measure for Scheduling. In Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms, pages 324–333, 2006.

[12] K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar. Paths, trees, and minimum latency tours. In Proc. 44th Symp. Foundations of Computer Science (FOCS), pages 36–45, 2003.

[13] B. Chazelle. *The Discrepancy Method: Randomness and Complexity.* Cambridge University Press, First Edition, 2000.

[14] C. Chekuri, S. Im, and B. Moseley. *Minimizing maximum response time and delay factor in broadcast scheduling.* In 17th Annual European Symposium on Algorithms (ESA), pages 444-455, 2009.

[15] C. Chekuri, S. Im, and B. Moseley. *Longest wait first for broadcast scheduling.* In Approximation and Online Algorithms, pages 62-74, 2010.

[16] DirecPC website, http://www.direcpc.com.

[17] J. Edmonds and K. Pruhs. *Broadcast scheduling: when fairness is fine.* Proc. of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 421-430, 2002.

[18] J. Edmonds and K. Pruhs. *A maiden analysis of Longest Wait First.* Proc. of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 818-827, 2004.

[19] T. Erlebach and A. Hall. *NP-hardness of broadcast scheduling and inapproximability of single-source unsplittable min-cost flow.* Proc. 13th ACM-SIAM Symp. on Disc. Algorithms, pages 194-202, 2002.

[20] R. Gandhi, S. Khuller, Y. Kim and Y-C. Wan. *Approximation algorithms for broadcast scheduling.* Proc. of the 9th Conference on Integer Programming and Combinatorial Optimization (IPCO), 2002.

[21] R. Gandhi, S. Khuller, S. Parthasarathy and A. Srinivasan. *Dependent Rounding in Bipartite Graphs*, Journal of the ACM 53 (2006), no. 3, 324–360, preliminary version in Proc. of the Forty-Third IEEE Symposium on Foundations of Computer Science (FOCS'02), pages 323-332, Nov. 2002.

[22] M. Goemans and J. Kleinberg, *An improved approximation ratio for the minimum latency problem*, Mathematical Programming, 82(1), pages 111–124, 1998.

[23] S. Im, and B. Moseley, *An online scalable algorithm for average flow time in broadcast scheduling.* ACM Transactions on Algorithms (TALG), 8(4), 39, 2012.

[24] B. Kalyanasundaram, K. Pruhs, and M. Velauthapillai, *Scheduling broadcasts in wireless networks*, Proc. of the 8th Annual European Symposium on Algorithms, pages 290–301, 2000.

[25] N. Karmarkar, and R. M. Karp, *An efficient approximation scheme for the one-dimensional bin-packing problem.* In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (pp. 312-320), 1982.

[26] S. Khuller and Y. Kim, *Equivalence of Two Linear Programming Relaxations for Broadcast Scheduling*, Operations Research Letters, 32(5), pages 473–478, 2004.

[27] S. Lovett, and R. Meka, *Constructive discrepancy minimization by walking on the edges*, In 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 61-67, 2012.

[28] A. Newman, O. Neiman and A. Nikolov, *Beck's Three Permutations Conjecture: A Counterexample and Some Consequences*, In 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 253-262, 2012.

[29] K. Pruhs, J. Sgall and E. Torng, *Online Scheduling.* Handbook of Scheduling: Algorithms, Models, and Performance Analysis, editor Joseph Y-T. Leung, CRC Press, 2004.

[30] J. Robert and N. Schabanel *Pull-Based Data Broadcast with Dependencies: Be Fair to Users, not to Items.* Proc. of the 18th ACM/SIAM Symp. on Discrete Algorithms (SODA), pages 238–247, 2007.

[31] T. Rothvoss, *Approximating Bin Packing within $O(\log OPT * \log\log OPT)$ bins.* Proc. of the Fifty-Fourth IEEE Symposium on Foundations of Computer Science (FOCS '13), to appear.

[32] J. Spencer, *Six standard deviations suffice.* Transactions of the American Mathematical Society, 289(2), 679-706, 1985.

A Proofs from Section 2

A.1 Defining the Blocks, and the Costs $C(S)$: Proof of Lemma 2.2

We begin with the following intuitive description of the original LP solution (x^*, y^*), and the fractional response time $\sigma(p,t)$. In this proof, we also present the details of how the blocks are constructed by the BCS algorithm, and present an intuitive explanation of the first phase of their algorithm.

Random Offset Rounding for a Request. One way to view $\sigma(p,t)$ is the following experiment (Lemma 1 in [4]). Say we choose α uniformly at random in $(0,1]$ and transmit page p at time the earliest time t_α where the fractional broadcast has accumulated α mass of page p from t, i.e., $t_\alpha = \arg\min_{t'} y^*(p,t,t') \geq \alpha$. Then the expected response time $E[t_\alpha - t]$ for $r(p,t)$ is equal to the LP cost for this request.

Random Offset Rounding for a Page. We now generalize the above intuition for getting a schedule for a page p over an interval of time, rather than satisfying just one request $r(p,t)$: To this end, consider a time interval $I = [t_1, t_2]$ such that $y^*(p,t_1,t_2) \geq 2$. Notice that $\mathsf{head}(I)$ and $\mathsf{tail}(I)$ are both non-empty intervals. We now obtain a broadcast sequence of page p as follows: Choose the *offset* α uniformly at random in $(0,1]$. For $i \in \mathbf{Z}_+$, define $t_{i+\alpha}$ to be the smallest $t' \leq t_2$ s.t $y^*(p,t_1,t') \geq i+\alpha$. If no such t' exists, set $t_{i+\alpha} = \infty$. Then, we transmit page p at times $t_{i+\alpha}$, and stop broadcasting when $t_{i+\alpha} = \infty$. The schedules for different offsets satisfy the following properties.

CLAIM 6. *Consider an interval $I = [t_1, t_2]$ s.t $y^*(p,t_1,t_2) \geq 2$. Define $frac(x) = x - \lfloor x \rfloor$ for $x \geq 0$. Then,*
(i) If $\alpha < frac(y^*(p,t_1,t_2))$, then the offset α makes $\lceil y^*(p,t_1,t_2) \rceil$ broadcasts within I. On the other hand, if $\alpha > frac(y^*(p,t_1,t_2))$, then it makes $\lfloor y^*(p,t_1,t_2) \rfloor$ broadcasts.
(ii) The broadcast schedules of two possible offsets α_1, α_2 are "interleaving": between two successive transmissions of α_1, there is exactly one transmission made by α_2.

Proof. The proof is quite straightforward. Indeed, suppose $\alpha < \text{frac}(y^*(p,t_1,t_2))$. Then notice that, for $i = \lfloor y^*(p,t_1,t_2) \rfloor$ is the largest index for which the quantity $\arg\min_{t'}(y^*(p,t_1,t') \geq i+\alpha)$ is at most t_2, and therefore $t_{i+\alpha}$ is finite. Therefore, the offset makes broadcasts for $i = 0, 1, \ldots, \lfloor y^*(p,t_1,t_2) \rfloor$, giving us the desired count. The case of larger α has a similar proof.

For the second part, suppose $\alpha_1 < \alpha_2$ WLOG. Then notice that for any $i \geq 0$, $\arg\min_{t'}(y^*(p,t_1,t') \geq i+\alpha_1) \leq \arg\min_{t'}(y^*(p,t_1,t') \geq i+\alpha_2)$, and also that $\arg\min_{t'}(y^*(p,t_1,t') \geq i+1+\alpha_1) > \arg\min_{t'}(y^*(p,t_1,t') \geq i+\alpha_2)$. The first inequality holds because $\alpha_1 < \alpha_2$, and the second because $\alpha_2 < 1$. Therefore, there is precisely one broadcast α_2 makes between two successive broadcast of α_1.

Furthermore, for the random schedule obtained, we can show (by applying the random offset argument) that the expected response time for all the requests that arrive in $\mathsf{head}(I)$ is equal to their total LP cost. Note that requests in $\mathsf{tail}(I)$, however may not even be satisfied by this schedule (for some choices of α).

The BCS Algorithm [4]. The main result of the first phase of [4] is that we can make the LP choose one offset integrally in each block, by allowing a small backlog. To achieve this, for each page p, it divides the time horizon $[1, T_{\max}+n]$ into disjoint intervals called *blocks* (which we will define later). Then, it solves an LP which picks one offset schedule for the corresponding page within each block, so as to (i) minimize the average response time, and (ii) ensure that the backlog is small. But what is the objective function of this LP? To this end, let us now try to see what the cost of choosing an offset α in block B for page p is: Indeed, by the random offset rounding above, the tentative schedule of this offset transmits at times $\{t_{i+\alpha}\}$. Therefore, each request arriving at time $t \in \mathsf{head}(B)$ has a fixed response time of $t_f - t$ where t_f is the smallest time after t when the page is transmitted by the offset.

But how do they handle the requests in $\mathsf{tail}(I)$ for each block? Such requests may not have any well-defined t_f within this block, and may only be satisfied by the next block for this page. This is in fact the main criterion for how BCS defines the blocks! Indeed, given a left end point t_1 for an interval, the right end point t_2 of the block $[t_1, t_2)$ is chosen in such a way that any request in $\mathsf{tail}([t_1,t_2))$ incurs a small cost, even if is "moved" to t_2, the starting time of the next block. The following definition helps to this end.

DEFINITION 3. (p-GOOD TIMESTEP) *Given a timestep t_1 and a page p, we say that a timestep t_2 is "p-good" if $\sigma(p,t_2) \leq 2\sigma(p,\tau)$ for all $\tau \in \mathsf{tail}([t_1,t_2])$.*

Moreover, they show (Lemma 2 in [4]) that a p-good timestep exists in any large enough interval.

LEMMA A.1. *Any time interval* $[t_1, t_2]$ *such that* $y^*(p, t_1, t_2) > \log T_{\max}$ *contains a p-good timestep.*

As a result, we have the following useful corollary.

COROLLARY A.1. *For any page p, and a block B for that page, the cumulative broadcast* $y^*(p, B) \leq O(\log T_{\max})$

For each page, BCS repeatedly uses the above lemma to partition the time horizon into disjoint intervals called *blocks*. Note that Corollary A.1 and Claim 6 together prove property (i) of Lemma 2.2. Claim 6 also proves property (ii) of Lemma 2.2.

By virtue of the way these blocks are chosen, we can intuitively "move" all requests in $\mathsf{tail}(I)$ to the beginning of the next interval. Formally we now define the cost of choosing a particular offset for a block. To this end, the cost $C(S)$ of a shift $S \equiv (B, \alpha)$ for a page p is calculated as follows:

(i) Let B_{-1} be the block corresponding to page p which is immediately before B, and consider the requests in its tail. Move all those requests to the starting time t_0 of block B. These requests now belong to B.

(ii) For any request for page p in B arriving at t (including the requests moved from the previous block), let t' denote the earliest time after t the page is broadcast according to the schedule defined by offset α, i.e., $\{t_{i+\alpha} : t_{i+\alpha} \in B\}$. Then, the request contributes $t' - t + 1$ to the cost $C(S)$. If no such time exists, then this request is satisfied in the next block, so we would have "moved" the request to the next block in step (i): in this case, include the moving cost $t_1 - t$ to $C(S)$ where t_1 is the ending time of the this block (which is also the starting time of the next block).

Property (iii) is proved in Bansal et al. [4], Lemma 5. It is easy to see that property (iv) follows from the way we have defined the cost $C(S)$ above. This completes the proof.

A.2 Adaptations of the Lovett Meka Partial Coloring Theorem We recall the actual theorem statement from [27], and then explain how our extra requirement can be handled. Indeed, Lovett and Meka show the following theorem.

THEOREM A.1. *[27]] Let* $\mathbf{y} \in [0,1]^m$ *be any starting point,* $\delta > 0$ *be an arbitrary error parameter,* $v_1, \ldots, v_n \in \mathbf{R}^n$ *vectors and* $\lambda_1, \ldots, \lambda_n \geq 0$ *parameters with*

$$\sum_{i=1}^{n} e^{-\lambda_i^2/16} \leq \frac{m}{16}. \tag{A.1}$$

Then there is a randomized $\widetilde{O}((m+n)^3/\delta^2)$*-time algorithm to compute a vector* $\mathbf{z} \in [0,1]^m$ *with*

(i) $z_j \in [0, \delta] \cup [1 - \delta, 1]$ *for at least half of the indices* $j \in [m]$,

(ii) $|v_i \cdot \mathbf{z} - v_i \cdot \mathbf{y}| \leq \lambda_i ||v_i||_2$, *for each* $i \in [n]$.

In order to incorporate our adaptation, we go into the details of their proof. In particular, their proof goes through unchanged all the way up to Claim 14. Here, they consider two cases, based on whether λ_j is small or large, and bound the small set by $m/16$. Here, in our case, the small set is bounded by $10m/16$ (by adding an extra $5m/8$). Therefore, the resulting bound in the claim becomes $12m/16$ instead of $m/4$.

The next change happens in Claim 16, where, if we replace the bound of $m/4$ from Claim 14 with $12m/16$, we get $m/16$ instead of $0.56m$. The rest of the proof is identical.