

# Online Disjoint Set Covers: Randomization Is Not Necessary

Marcin Bienkowski  

University of Wrocław, Poland

Jarosław Byrka  

University of Wrocław, Poland

Łukasz Jeż  

University of Wrocław, Poland

---

## Abstract

In the online disjoint set covers problem, the edges of a hypergraph are revealed online, and the goal is to partition them into a maximum number of disjoint set covers. That is,  $n$  nodes of a hypergraph are given at the beginning, and then a sequence of hyperedges (subsets of  $[n]$ ) is presented to an algorithm. For each hyperedge, an online algorithm must assign a color (an integer). Once an input terminates, the gain of the algorithm is the number of colors that correspond to valid set covers (i.e., the union of hyperedges that have that color contains all  $n$  nodes).

We present a deterministic online algorithm that is  $O(\log^2 n)$ -competitive, exponentially improving on the previous bound of  $O(n)$  and matching the performance of the best randomized algorithm by Emek et al. [ESA 2019].

For color selection, our algorithm uses a novel potential function, which can be seen as an online counterpart of the derandomization method of conditional probabilities and pessimistic estimators. There are only a few cases where derandomization has been successfully used in the field of online algorithms. In contrast to previous approaches, our result extends to the following new challenges: (i) the potential function derandomizes not only the Chernoff bound, but also the coupon collector's problem, (ii) the value of OPT of the maximization problem is not bounded a priori, and (iii) we do not produce a fractional solution first, but work directly on the input.

**2012 ACM Subject Classification** Theory of computation → Online algorithms

**Keywords and phrases** Disjoint Set Covers, Derandomization, pessimistic Estimator, potential Function, online Algorithms, competitive Analysis

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2025.18

**Funding** Supported by Polish National Science Centre grants 2022/45/B/ST6/00559, 2020/39/B/ST6/01641, and 2020/39/B/ST6/01679.

**Acknowledgements** The authors are grateful to the anonymous reviewers for their insightful comments.

## 1 Introduction

In this paper, we study online algorithms for maximizing the number of set covers of a set of nodes. We focus on a hypergraph (set system)  $G = (V, E)$  that has  $n = |V|$  nodes and where each hyperedge  $S \in E$  is a subset of nodes from  $V$ . A subset  $E' \subseteq E$  is called *set cover* if  $\bigcup_{S \in E'} S = V$ , i.e., every node is covered by at least one hyperedge of  $E'$ . In the *disjoint set covers* (DSC) problem [15, 12, 20], the goal is to partition the set of hyperedges  $E$  into *maximum number* of mutually disjoint subsets  $E = E_1 \uplus E_2 \uplus \dots \uplus E_k$ , where each  $E_j$  is a set cover. Note that  $E$  is a multi-set, i.e., it can contain multiple copies of the same hyperedge.

The problem has been studied in a theoretical setting, but as we discuss later, it also finds applications in sensor networks [12] or assignment tasks [20].



© Marcin Bienkowski, Jarosław Byrka, and Łukasz Jeż;  
licensed under Creative Commons License CC-BY 4.0

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).

Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thăng;

Article No. 18; pp. 18:1–18:16



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



**Coloring Perspective.** When constructing a solution to the DSC problem, it is convenient to think in terms of coloring hyperedges.<sup>1</sup> Each color then corresponds to a subset of hyperedges that have that color, and the color is *fully used* if the hyperedges colored with it form a set cover. The problem then becomes equivalent to coloring hyperedges, so that the number of fully used colors is maximized.

**Online Variant.** In this paper, we focus on the *online* variant of the DSC problem, where the set  $V$  is given in advance, but the hyperedges of  $E$  arrive in an online fashion. Once a hyperedge  $S \in E$  arrives, it must be colored immediately and irrevocably. Again, the goal is to maximize the number of fully used colors. The performance of an online algorithm is measured by the competitive ratio, i.e., the ratio of the number of fully used colors produced by the optimal *offline* algorithm  $\text{OPT}$  to that of an online algorithm  $\text{ALG}$ .

**Our Contribution.** We present a deterministic online  $O(\log^2 n)$ -competitive algorithm  $\text{DET}$  for the DSC problem, which exponentially improves on the  $O(n)$ -competitive algorithm by Emek et al. [14] and matches the performance of their randomized algorithm [14]. We discuss the challenges and technical contribution in more detail in Subsection 1.3.

## 1.1 Offline Scenario: Previous Results

The disjoint set covers problem is a fundamental NP-complete problem [12] that can be approximated within a factor of  $(1 + o(1)) \cdot \ln |V|$  [20] and cannot be approximated within a factor of  $(1 - \varepsilon) \cdot \ln |V|$  for  $\varepsilon > 0$  unless  $\text{NP} \subseteq \text{DTIME}(n^{\log \log n})$  [15].<sup>2</sup>

**OPT vs. Min-degree.** We use  $\text{OPT}(E)$  to denote the maximum number of disjoint set covers of a hypergraph  $G = (V, E)$ . This value is also called *cover-decomposition number* [6]. We denote the minimum degree of the hypergraph  $G = (V, E)$  as  $\delta(E) \triangleq \min_{i \in V} |S \in E : i \in S|$ . Note that trivially  $\text{OPT}(E) \leq \delta(E)$ . While this bound may not be tight (cf. Subsection 1.4),  $\delta(E)$  serves as a natural benchmark for approximation and online algorithms.

**Random Coloring and its Straightforward Derandomization.** An  $O(\log n)$ -approximation algorithm for the *offline* DSC problem [21] can be obtained by coloring each hyperedge with a color chosen uniformly at random from the palette of  $\Theta(\delta(E)/\log n)$  colors. To analyze this approach, we focus on a single node  $i \in V$ . We say that node  $i$  *gathers* color  $r$  if  $i$  is contained in a hyperedge colored with  $r$ . Since node  $i$  is contained in at least  $\delta(E)$  hyperedges, and there are  $\Theta(\delta(E)/\log n)$  available colors,  $i$  gathers all palette colors with high probability. By the union bound, this property holds for all nodes, i.e., all  $\Theta(\delta(E)/\log n)$  colors are fully used by an algorithm (also with high probability). Since  $\text{OPT}(E) \leq \delta(E)$ , the approximation ratio of  $O(\log n)$  follows.

The hyperedges can be processed in a fixed order, and the random choice of a color can be replaced by the deterministic one by a simple application of the method of conditional probabilities [21, 3].

<sup>1</sup> The DSC problem should not be confused with the hyperedge coloring problem, which requires that the hyperedges of the same color be disjoint.

<sup>2</sup> The authors of [15] call this problem *set cover packing* or *one-sided domatic problem*.

## 1.2 Online Scenario: Previous Results

In the online case, an algorithm first learns the set of nodes  $V$ , and then the edges of  $E$  are revealed sequentially. For notational convenience, we use  $E$  to denote both the set and a sequence of hyperedges (the input). We use  $\text{ALG}(E)$  to denote the number of fully used colors in the solution of an online algorithm  $\text{ALG}$ .

It is important to note that no parameter of the hypergraph other than the number of nodes is known a priori. In particular, an online algorithm does not know the value of  $\delta(E)$  in advance.

**Competitive Ratio.** We say that  $\text{ALG}$  is (*non-strictly*)  $c$ -competitive if there exists  $\beta \geq 0$  such that

$$\text{ALG}(E) \geq \text{OPT}(E) / c - \beta. \quad (1)$$

While  $\beta$  can be a function of  $n$ , it cannot depend on the input sequence  $E$ . If (1) holds with  $\beta = 0$ , the algorithm is *strictly*  $c$ -competitive.

For randomized algorithms, we replace  $\text{ALG}(E)$  with its expected value taken over the random choices of the algorithm.

**Randomized Algorithms.** The current best randomized algorithm was given by Emek et al. [14]; it achieves the (strict) competitive ratio of  $O(\log^2 n)$ . The general idea of their algorithm is as follows. To color a hyperedge  $S$ , their algorithm first computes the minimum degree  $\delta^*$  of a node from  $S$ . By a combinatorial argument, they show that they can temporarily assume that  $\delta(E) = O(n \cdot \delta^*)$ . Their algorithm then chooses  $\ell$  uniformly at random from the set  $\{\delta^*, 2\delta^*, 4\delta^*, \dots, 2^{\log n} \cdot \delta^*\}$ ; with probability  $\Omega(1/\log n)$  such  $\ell$  is a 2-approximation of  $\delta(E)$ . Finally, to color  $S$ , it chooses a color uniformly at random from the palette of  $\Theta(\ell/\log^2 n)$  colors, using the arguments similar to those in the offline case.

We refrain from discussing it further here, as we present a variant of their algorithm, called  $\text{RAND}$ , in Subsection 2.1 (along with a description of the differences from their algorithm).

The best lower bound for the (strict and non-strict) competitive ratio of a randomized algorithm is  $\Omega(\log n / \log \log n)$  [14]; it improves on an earlier bound of  $\Omega(\sqrt{\log n})$  by Pananjady et al. [20].

**Deterministic Algorithms.** The deterministic case is well understood if we restrict our attention to *strict* competitive ratios. In such a case the lower bound is  $\Omega(n)$  [20]. The asymptotically matching upper bound  $O(n)$  is achieved by a simple greedy algorithm [14].

On the other hand, the current best lower bound for the non-strict deterministic competitive ratio is  $\Omega(\log n / \log \log n)$  [14].<sup>3</sup> The  $O(n)$  upper bound of [14] clearly holds also in the non-strict setting, but no better algorithm has been known so far.

**Lack of General Derandomization Tools.** Unlike approximation algorithms, derandomization is extremely rare in online algorithms.<sup>4</sup> To understand the difficulty, consider the standard (offline) method of conditional probabilities [3] applied to the offline DSC problem:

<sup>3</sup> This discrepancy in achievable strict and non-strict competitive ratios is quite common for many maximization problems: the non-strict competitive ratio allows the algorithm to have zero gain on very short sequences, thus avoiding initial choices that would be bad in the long run.

<sup>4</sup> Many online problems (e.g., caching [23, 1] or metrical task systems [7, 8, 9]) exhibit a provable exponential discrepancy between the performance of the best randomized and deterministic algorithms. For many other problems, the best known deterministic algorithms are quite different (and more complex) than randomized ones.

■ **Table 1** Previous and new bounds on the strict and non-strict competitive ratios for the online DSC problem, for randomized and deterministic algorithms.

	Upper bound	Lower bound
randomized strict and non-strict	$O(\log^2 n)$ [14]	$\Omega(\log n / \log \log n)$ [14]
deterministic strict	$O(n)$ [14]	$\Omega(n)$ [20]
deterministic non-strict	$O(n)$ [14] <b><math>O(\log^2 n)</math></b> (Theorem 5)	$\Omega(\log n / \log \log n)$ [14]

the resulting algorithm considers all the random choices (color assignments) it could make for a given hyperedge  $S$ , and computes the probability that future random choices, *conditioned on the current one*, will lead to the desired solution. Choosing the color that maximizes this probability ensures that the probability of reaching the desired solution does not decrease as subsequent hyperedges are processed. In some cases, exact computations are not possible, but the algorithm can instead estimate this probability by computing a so-called *pessimistic estimator* [22]. This is not feasible in the online setting, since an algorithm does not know the future hyperedges, and thus cannot even estimate these probabilities.<sup>5</sup>

### 1.3 Our Technical Contribution

In this paper, we present a deterministic online polynomial-time algorithm DET that is  $O(\log^2 n)$ -competitive, which exponentially improves the previous bound of  $O(n)$ . This resolves an open question posed by the authors of [14] who asked whether the method of conditional expectations could be used to derandomize their algorithm.

Our bound is obtained for the non-strict competitive ratio: as we discussed in the previous subsection, this is unavoidable for the DSC problem. Our result requires a relatively small ( $1/4$ ) additive term  $\beta$  in the definition of the competitive ratio (1).

We begin by constructing a *randomized* solution RAND. It will be a variation of the approach by Emek et al. [14]; we present it in Subsection 2.1. As RAND is closely related to the previous randomized algorithm, it is quite plausible that it achieves the same competitive ratio of  $O(\log^2 n)$ . However, our analysis does not support this conjecture. Instead, we use RAND as a stepping stone to our deterministic algorithm in the following way.

- We define a particular random event, denoted  $\mathcal{T}_E$ , of RAND executed on instance  $E$ .
- We show that for each execution of RAND satisfying  $\mathcal{T}_E$ , it holds that  $\text{RAND}(E) \geq \Omega(1/\log^2 n) \cdot \text{OPT}(E) - 1/4$ .

We will provide the exact definition of the event  $\mathcal{T}_E$  in Subsection 2.2. For now, we just note that  $\mathcal{T}_E$  is a property that certifies that, *at each step  $t$* , we can relate the number of colors gathered so far by each node  $i$  to its current degree.

While a relaxed version of the property  $\mathcal{T}_E$  (requiring that the relation between the gathered colors and the current degree *only at the end of the input*) follows quite easily with a constant probability, it seems that  $\mathcal{T}_E$  itself is quite strong and does not hold with a reasonable probability. However, this is not an obstacle to creating a deterministic solution, as we show in the following claim.

<sup>5</sup> As observed by Pananjady et al. [20], however, these probabilities can be estimated if an online algorithm knows the final min-degree  $\delta(E)$  in advance, which would lead to an  $O(\log n)$ -competitive algorithm for this semi-offline scenario.

- It is possible to replace the random choices of RAND by deterministic ones (in an online manner), so that  $\mathcal{T}_E$  always holds.

This will immediately imply the competitive ratio of the resulting deterministic algorithm.

Our approach is based on a novel potential function that guides the choice of colors. As we discuss later in Subsection 2.2, our approach can be seen as an online-computable counterpart of the method of conditional probabilities that simultaneously controls the derandomization of the Chernoff bound and the coupon collector's problem.

## 1.4 Related Work

Applications of the DSC problem include allocating servers to users in file systems and users to tasks in crowd-sourcing platforms [20].

Another application arises in a sensor network, where each node corresponds to a monitoring target and each hyperedge corresponds to a sensor that can monitor a set of targets. At any given time, all targets must be monitored. A possible battery-saving strategy is to partition the sensors into disjoint groups (each group covering all targets) and activate only one group at a time, while the other sensors remain in a low-power mode [12].

When the assumption that each sensor can only participate in a single group is dropped, this leads to more general sensor coverage problems. The goal is then to maximize the lifetime of the network of sensors while maintaining the coverage of all targets. The offline variants of this problem have been studied both in general graphs [4, 13, 21] and in geometric settings [11, 16].

Another line of work studied the relationship between the minimum degree  $\delta(E)$  and the cover-decomposition number  $\text{OPT}(E)$ . As pointed out in Subsection 1.1,  $\delta(E)/\text{OPT}(E) \geq 1$ . This ratio is constant if  $G$  is a graph [17], and it is at most  $O(\log n)$  for every hypergraph; the latter bound is asymptotically tight [6]. Interestingly enough, all the known papers on the DSC problem (including ours) relate the number of disjoint set covers to  $\delta(E)$ . It is an open question whether our estimate of the competitive ratio is tight: it could possibly be improved by relating the gain of an algorithm directly to  $\text{OPT}(E)$  instead of  $\delta(E)$ .

## 1.5 Preliminaries

An input to our problem is a gradually revealed hypergraph  $G = (V, E)$ .  $V$  consists of  $n$  nodes numbered from 1 to  $n$ , i.e.,  $V = [n]$ . The set  $E$  of hyperedges is presented one by one: in a step  $t$ , an algorithm learns  $S_t \in E$ , where  $S_t \subseteq [n]$ , and has to color it. We say that node  $i$  *gathers* color  $r$  if  $i$  is contained in a hyperedge colored with  $r$ . We say that a color  $r$  is *fully used* if all nodes have gathered it. The objective of an algorithm is to maximize the number of fully used colors.

At any point in time, the *degree* of a node  $j$ , denoted  $\text{deg}(j)$ , is the current number of hyperedges containing  $j$ . Let  $\delta(E) = \min_{i \in [n]} |\{S_t \in E : v_i \in S_t\}|$ , i.e.,  $\delta(E)$  is the minimum degree of  $G$  at the end of the input  $E$ . Clearly,  $\text{OPT}(E) \leq \delta(E)$ . It is important to note that  $\delta(E)$  is not known in advance to an online algorithm.

■ **Algorithm 1** Definition of RAND for a hyperedge  $S$  in step  $t$ .

---

```

1: ▷ Initialization
2: for each node  $i \in [n]$  do
3:    $p(i) \leftarrow 0$                                      ▷ all nodes start in phase 0
4:   for each integer  $k \geq 0$  do                         ▷ and they have no colors yet
5:      $C_{i,k} \leftarrow \emptyset$ 

6: ▷ Runtime
7: for each hyperedge  $S$  appearing in sequence  $E$  do
8:    $p_S \leftarrow \min_{i \in S} p(i)$ 
9:   Sample  $k^*$  from  $\{p_S, p_S + 1, \dots, p_S + h - 1\}$    ▷ uniform distribution
10:  Sample color  $r$  from  $R_{k^*}$                                ▷ uniform distribution
11:  for each node  $i \in S$  do
12:     $C_{i,k^*} \leftarrow C_{i,k^*} \cup \{r\}$ 
13:    if  $|C_{i,p(i)}| \geq q_{p(i)}$  then                       ▷ end node phase if it gathered  $q_{p(i)}$  colors
14:       $p(i) \leftarrow p(i) + 1$ 

```

---

## 2 Our Algorithm

### 2.1 Definition of RAND

We start with some notions, defined for each integer  $k \geq 0$ :

$$h \triangleq \lceil \log n \rceil, \quad q_k \triangleq \left\lceil \left(1 - \frac{1}{2n}\right) \cdot 2^k \right\rceil, \quad R_k \triangleq \{2^k, \dots, 2^{k+1} - 1\}.$$

Note that  $|R_k| = 2^k$ .

For each node  $i$  independently, RAND maintains its *phase*  $p(i)$ , initially set to 0. We use  $C_{i,k}$  to denote the set of colors from the palette  $R_k$  that node  $i$  has gathered so far. A phase  $k$  for node  $i$  ends when it has gathered  $q_k$  colors from palette  $R_k$ . In such a case, node  $i$  increments its phase number  $p(i)$  at the end of the step.

We now describe the behavior of RAND in a single step, when a hyperedge  $S$  appears. Let  $p_S = \min_{i \in S} p(i)$ , where  $p(i)$  is the phase of node  $i$  before  $S$  appears. RAND first chooses a random integer  $k^*$  uniformly from the set  $\{p_S, p_S + 1, \dots, p_S + h - 1\}$ . Second, RAND chooses a color  $r$  uniformly at random from the set  $R_{k^*}$  and colors  $S$  with it; in effect all nodes in  $S$  gather color  $r$ .

The pseudocode of RAND is given in Algorithm 1.

**Comparison with the Previous Randomized Algorithm.** RAND is closely related to the randomized algorithm by Emek et al. [14]. The main difference is that the phases of the nodes in their algorithm are of fixed lengths, being powers of 2.<sup>6</sup> They use probabilistic arguments to argue that with high probability each node gathers  $q_k$  colors in a phase of length  $\Theta(2^k \cdot \log^2 n)$ . Instead, we treat the number of colors gathered in a phase as a hard constraint (we require that each node gathers  $q_k$  of them), and instead the phase lengths of the nodes become random variables. As it turns out, this subtle difference allows us to derandomize the algorithm.

<sup>6</sup> The pseudocode of their algorithm does not use phases, but with some fiddling with constants, it can be transformed into one that does.

Another small difference concerns the color selection. While RAND chooses the color uniformly at random from the set  $R_{k^*}$ , the algorithm of [14] would choose it from the set  $\uplus_{j=0}^{k^*} R_j$ . This difference affects only the constant factors of the analysis.

**Gain of RAND.** As mentioned earlier, the gain of RAND is directly ensured by the algorithm definition *provided* that each node has completed some number of phases.

Note that  $q_k$  is slightly less than  $|R_k| = 2^k$ ; this gives a better bound on the the expected phase lengths, while still ensuring that if all nodes completed phase  $\ell$ , they gathered at least  $2^{\ell-1}$  common colors.

► **Lemma 1.** *If every node has completed phase  $\ell$ , then  $RAND(E) \geq 2^{\ell-1}$ .*

**Proof.** In phase  $\ell$ , each node gathers at least  $q_\ell$  colors from the palette  $R_\ell$ , i.e., all colors from  $R_\ell$  except at most  $|R_\ell| - q_\ell \leq 2^\ell / (2n)$  colors. Thus, all nodes share at least  $|R_\ell| - n \cdot (2^\ell / (2n)) = 2^{\ell-1}$  common colors from  $R_\ell$ , i.e., RAND fully uses at least  $2^{\ell-1}$  colors. ◀

Note that we could sum the above bound over all phases completed by all nodes, but this would not change the asymptotic gain.

The above lemma points us to the goal: to show that for an input  $E$  with a sufficiently large  $\delta(E)$ , each node completes an appropriately large number of phases. In Subsection 2.3, we show that if  $\delta(E) = \Omega(2^\ell \cdot \log^2 n)$ , then each node completes  $\ell$  phases, *provided* a certain random event  $\mathcal{T}_E$  occurs.

## 2.2 Constructing the Potential: Insights and Definitions

In the field of online algorithms, the derandomization has been successfully conveyed several times by replacing the method of conditional probabilities by an *online-computable potential function* that guides the choice of the deterministic algorithm [2, 5, 10, 18].

This method is based on the following framework. First, introduce  $\ell$  random expressions  $\{Z_i\}_{i=1}^\ell$  to be controlled. Second, define a potential function  $\Phi = \sum_{i=1}^\ell \exp(Z_i)$ . Third, show that the random actions of an algorithm at each step decrease  $\exp(Z_i)$  (for each  $i$ ) in expectation, which implies that  $\Phi$  decreases as well. (By the probabilistic method, this implies the existence of a *deterministic action* of an algorithm in a step that does not increase  $\Phi$ .) Finally, by the non-negativity of the exponential function,  $\exp(Z_i)$  is bounded by the initial value  $\Phi^0$  of the potential (in each step), which shows that  $Z_i$  can always be bounded by  $\ln(\Phi^0)$ . This process can be seen as a derandomization of the Chernoff bound / high probability argument.

In order to apply this very general framework, we must overcome several technical difficulties, which we explain below. Except for the definition of  $\Phi$  (and related definitions of  $w_{ik}$ ,  $c_{ik}$ ,  $d_k$  and  $Z_i$ ), the discussion in this section is informal and will not be used in formal proofs later.

**Variables to be Controlled.** The first step is to identify the variables to control. Natural choices are the node degrees and the number of colors gathered so far by each node.

For this purpose, we define  $c_{ik} = |C_{ik}|$  for each node  $i$  and each phase  $k \geq 0$ .

We also introduce counters  $w_{ik}$ , which are initially set to zero. Recall that whenever a new hyperedge  $S$  containing node  $i$  arrives, RAND computes  $p_S = \min_{i \in S} p(i)$ . For each node  $i \in S$  such that  $p(i) \leq p_S + h - 1$ , we increment the counter  $w_{i,p(i)}$ . Note that these  $w_{i,p(i)}$  variables are incremented exactly for nodes  $i$  for which there is a non-zero probability of increasing their set of colors  $C_{i,p(i)}$  (as then a random integer  $k^*$  chosen by RAND has a non-zero chance of being equal to  $p(i)$ ).

By the definition of  $w_{ik}$ , at each step  $\sum_{k \geq 0} w_{ik} \leq \deg(i)$ . While these quantities are not necessarily equal, we will treat  $\sum_{k \geq 0} w_{ik}$  as a good proxy for  $\deg(i)$  and deal with the discrepancy between these two quantities later.

**Linking the Variables.** Now we want to introduce an expression that links  $\sum_{k \geq 0} w_{ik}$  (the proxy for degree) with  $\sum_{k \geq 0} c_{ik}$  (the number of colors gathered) for a node  $i$ . A simple difference of these two terms does not make sense: the expected growth of  $c_{ik}$  varies over time, since it is easier to gather new colors when  $c_{ik}$  is small. This effect has been studied in the context of the coupon collector's problem [19]. To overcome this issue, we introduce the following function, defined for each integer  $k \geq 0$ :

$$d_k(m) \triangleq h \cdot \sum_{j=1}^m \frac{2^k}{2^k - j + 1} \quad (\text{defined for each } m \leq 2^k)$$

Note that in a process of choosing random colors from palette  $R_k$  of  $2^k$  colors, the expected number of steps till  $m$  different colors are gathered is  $d_k(m)/h$ .

Now we focus on a single node  $i$  in phase  $p(i)$ . We consider a sequence of hyperedges  $S$  containing node  $i$ , neglecting those hyperedges  $S$  for which  $p(i) \geq p_S + h$ . That is, all considered hyperedges increment the counter  $w_{i,p(i)}$ . Then, the value of  $d_k(c_{i,p(i)})$  corresponds to the expected number of such hyperedges needed to gather  $c_{i,p(i)}$  colors from the palette  $R_{p(i)}$ . We can thus use the expression  $\sum_{k \geq 0} (w_{ik} - 2 \cdot d_k(c_{ik}))$  to measure the progress of node  $i$ : small values of this expression indicate that it is gathering colors quite fast, while large values indicate that it is falling behind. Note that since  $c_{ik} \leq q_k \leq 2^k$ , the value of  $d_k(c_{ik})$  is always well defined.

We note that the previous applications of the potential function method [2, 5, 10, 18] did not require such transformations of variables: in their case, the potential function was used to guide deterministic *rounding*: the function  $\Phi$  directly compared the cost (or gain) of a deterministic algorithm with that of an online fractional solution. Instead, our solution operates directly on the input, without the need to generate a fractional solution first.

**Scaling.** Using the random choices of RAND, we can argue that in expectation the value of  $Z_i^* \triangleq \sum_{k \geq 0} (w_{ik} - 2 \cdot d_k(c_{ik}))$  is decreasing in time. However, to argue that  $\exp(Z_i^*)$  is also decreasing in expectation, we would have to ensure that  $Z_i^*$  is upper-bounded by a small constant (and use the fact that  $\exp(x)$  can be approximated by a linear function for small  $x$ ).

In the previous papers [2, 5, 10, 18], this property was achieved by scaling down  $Z^*$  by the value of OPT. An algorithm was then either given an upper bound on OPT (in the case of the throughput maximization of the virtual circuit routing [10]) or OPT was estimated by standard doubling techniques (in the case of the cost minimization for set cover variants [2, 5, 18]). In the latter case, the algorithm was restarted each time the estimate on OPT was doubled. Unfortunately, the DSC problem (which is an unbounded-gain maximization problem) does not fall into any of the above categories, and the doubling approach does not seem to work here.

Instead, we replace the scaling with a weighted average. That is, for each node  $i$ , we define

$$Z_i \triangleq \sum_{k \geq 0} \frac{w_{ik} - 2 \cdot d_k(c_{ik})}{4h \cdot 2^k}$$

and the potential as

$$\Phi \triangleq \sum_{i \in [n]} \exp(Z_i).$$



Note that all counters and variables defined above are random variables; they depend on particular random choices of RAND. We use  $p^t(i)$ ,  $\deg^t(i)$ ,  $w_{ik}^t$ ,  $c_{ik}^t$ ,  $Z_i^t$  and  $\Phi^t$  to denote the values of the corresponding variables at the end of step  $t$  of the algorithm (after RAND has processed the hyperedge presented in step  $t$ ). The value of  $t = 0$  corresponds to the state of these variables at the beginning of the algorithm; note that  $p^0(i) = \deg^0(i) = w_{ik}^0 = c_{ik}^0 = Z_i^0 = 0$  for all  $i$  and  $k$ . Therefore,

$$\Phi^0 = n. \tag{2}$$

**Random event  $\mathcal{T}_E$ .** For an input instance  $E$  consisting of  $T$  steps, we define a random event  $\mathcal{T}_E$  that occurs if  $\Phi^t \leq n$  for each step  $t \in \{0, \dots, T\}$  of RAND execution on input  $E$ .

### 2.3 Relating Potential to Algorithm Performance

We begin by presenting the usefulness of the event  $\mathcal{T}_E$ . We emphasize that the following lemma holds for all executions of RAND in which the event  $\mathcal{T}_E$  occurs. Its proof is deferred to Section 3.

► **Lemma 2.** *Fix a sequence  $E$  such that  $\delta(E) > 24h \cdot \ln(4e \cdot n) \cdot 2^\ell$  for some integer  $\ell \geq 0$ . If  $\mathcal{T}_E$  occurs, then each node has completed its phase  $\ell$ .*

Using the lemma above, we can relate the gain of RAND on an arbitrary input  $E$  to that of OPT, if only  $\mathcal{T}_E$  occurs.

► **Lemma 3.** *Fix a sequence  $E$ . If  $\mathcal{T}_E$  occurs, then  $\text{RAND}(E) \geq \text{OPT}(E)/(96h \cdot \ln(4e \cdot n)) - 1/4$ .*

**Proof.** Let  $r \triangleq 24h \cdot \ln(4e \cdot n)$ . We consider two cases.

- First, we assume that  $\delta(E) > r$ . Then we can find an integer  $\ell \geq 0$  such that  $r \cdot 2^\ell < \delta(E) \leq r \cdot 2^{\ell+1}$ . By Lemma 2, each node then completes its phase  $\ell$ , and so by Lemma 1,  $\text{RAND}(E) \geq 2^{\ell-1} \geq \delta(E)/(4r)$ .
- Second, we assume that  $\delta(E) \leq r$ . Trivially,  $\text{RAND}(E) \geq 0 \geq (\delta(E) - r)/(4r)$ .

In both cases,  $\text{RAND}(E) \geq (\delta(E) - r)/(4r) \geq \text{OPT}(E)/(4r) - 1/4$ . ◀

### 2.4 Derandomization of RAND

To analyze the evolution of  $\Phi$ , we note that  $\Phi^t$  (and also other variables  $w_{ik}^t$ ,  $c_{ik}^t$  or  $Z_i^t$ ) depends only on the random choices of RAND till step  $t$  (inclusively). Moreover,  $\{\Phi^t\}_{t \geq 0}$  is a supermartingale with respect to the random choices of RAND in consecutive steps. Specifically, the following lemma holds; its proof is deferred to Section 4.

► **Lemma 4.** *Fix a step  $t$  and an outcome of random choices till step  $t - 1$  inclusively. (In particular, this will fix the value of  $\Phi^{t-1}$ .) Then,  $\mathbb{E}[\Phi^t] \leq \Phi^{t-1}$ , where the expectation is taken exclusively over random choices of RAND in step  $t$ .*

The above lemma states that the value of  $\Phi$  is non-increasing in expectation. In fact, an inductive application of this lemma shows that  $\mathbb{E}[\Phi^t] \leq \Phi^0 = n$ . However, this does not imply that  $\mathcal{T}_E$  occurs with a reasonable probability, especially when the input length is large.<sup>7</sup>

<sup>7</sup> By Markov's inequality, for a chosen step  $t$ ,  $\Phi^t \leq 2n$  holds with probability at least  $1/2$ . While such a relaxed bound on  $\Phi^t$  would be sufficient for our needs, in our proof, we need such a bound to hold for all steps  $t$  simultaneously.

## 18:10 Online Disjoint Set Covers: Randomization Is Not Necessary

However, using Lemma 4, we can easily derandomize RAND using the method of conditional probabilities using potential  $\Phi$  as an online-computable counterpart of a pessimistic estimator. To do this, we proceed iteratively, ensuring at each step  $t$  that  $\Phi^t \leq n$ . This is trivial at the beginning, since  $\Phi^0 = n$  by (2).

Suppose we have already fixed the random choices of RAND till step  $t - 1$  inclusively and have  $\Phi^{t-1} \leq n$ . Consider a hyperedge  $S$  presented in step  $t$ . Note that all other variables indexed by  $t - 1$ , such as  $p^{t-1}(i)$ , are also fixed. Then Lemma 4 states that  $\mathbb{E}[\Phi^t] \leq \Phi^{t-1} \leq n$ . That is, the random choice of a color at step  $t$  guarantees that  $\mathbb{E}[\Phi^t] \leq n$ . This choice is made from a finite and well-defined set of colors  $\mathcal{R} \triangleq \biguplus_{p_S \leq k \leq p_S + h - 1} R_k$ , where  $p_S = \min_{i \in S} p^{t-1}(i)$ .

By the probabilistic method, at each step  $t$ , there exists a *deterministic* choice of a color from  $\mathcal{R}$  that ensures that  $\Phi^t \leq n$ . The resulting algorithm is called DET. (If more than one color leads to  $\Phi^t \leq n$ , DET chooses any of them.) Since  $|\mathcal{R}|$  is bounded by a polynomial of  $n$  and  $|E|$ , DET can be implemented in polynomial time by simply checking all possible colors of  $\mathcal{R}$ .

► **Theorem 5.** *DET is  $O(\log^2 n)$ -competitive for the DSC problem.*

**Proof.** As described above, DET guarantees that  $\Phi^t \leq n$  for each step  $t \in \{0, \dots, T\}$ , i.e.,  $\mathcal{T}_E$  occurs. Note that Lemma 3 lower-bounds the gain of RAND in every execution conditioned on  $\mathcal{T}_E$ , and DET can be seen as such an execution. Therefore, the bound of Lemma 3 can be applied, yielding

$$\text{DET}(E) \geq \frac{\text{OPT}(E)}{96 \cdot h \cdot \ln(4e \cdot n)} - \frac{1}{4},$$

i.e., the competitive ratio of DET is at most  $96 \cdot h \cdot \ln(4e \cdot n) = O(\log^2 n)$ .

Note that, by the lower bounds of [20, 14], an additive term (in our case equal to  $1/4$ ) is inevitable for obtaining a sub-linear competitive ratio. ◀

### 3 Bounding Number of Phases

In this section, we fix an input sequence  $E$  consisting of  $T$  steps. Our goal is to estimate the number of phases of nodes in the execution of RAND, conditioned on the random event  $\mathcal{T}_E$ , i.e., to prove Lemma 2. To this end, we consider a node  $i$ , assume that it has completed  $\ell$  phases, and show an upper bound on the final degree of  $i$  as a function of  $\ell$ .

**Bounding Variables w Using Potential.** Recall that in some steps where the degree of a node  $i$  grows, the counter  $w_{i,p(i)}$  is incremented. Thus, our first goal is to upper-bound values of these counters at the end of the execution of RAND.

Below,  $H(m)$  denotes the  $m$ -th harmonic number. The following technical claim is proved in Appendix A.

▷ **Claim 6.** For each  $k \geq 0$  it holds that  $H(2^k) - H(2^k - q_k) \leq \ln(4e \cdot n)$ .

► **Lemma 7.** Fix a step  $t \leq T$ , a node  $i$  and a phase  $k \geq 0$ . Then,  $d_k(c_{ik}^t) \leq h \cdot \ln(4e \cdot n) \cdot 2^k$ .

**Proof.** Note that at all times,  $c_{ik}^t \leq q_k$ . Using the definition of  $d_k$ ,

$$\begin{aligned} d_k(c_{ik}^t) &\leq d_k(q_k) = h \cdot \sum_{j=1}^{q_k} \frac{2^k}{2^k - j + 1} \\ &= h \cdot (H(2^k) - H(2^k - q_k)) \cdot 2^k \\ &\leq h \cdot \ln(4e \cdot n) \cdot 2^k. \end{aligned} \quad \text{(by Claim 6)} \quad \blacktriangleleft$$

► **Lemma 8.** Fix a node  $i$  and a phase  $\ell \geq 0$ . If the event  $\mathcal{T}_E$  occurs, then  $\sum_{k=0}^{\ell} w_{ik}^T \leq 8h \cdot \ln(4e \cdot n) \cdot 2^\ell$ .

**Proof.** Fix the last step  $t \leq T$  in which  $\sum_{k=0}^{\ell} w_{ik}$  increases. By the choice of  $t$ , we have  $w_{ik}^t = c_{ik}^t = 0$  for every phase  $k > \ell$ .

Since  $\mathcal{T}_E$  occurs,  $n \geq \Phi^t = \sum_{j \in [n]} \exp(Z_j^t)$ . Due to the non-negativity and monotonicity of the exponential function,  $Z_i^t \leq \ln n$ . Using the definition of  $Z_i^t$ , we get

$$\begin{aligned} \ln n \geq Z_i^t &= \sum_{k \geq 0} \frac{w_{ik}^t - 2 \cdot d_k(c_{ik}^t)}{4h \cdot 2^k} \\ &= \sum_{k=0}^{\ell} \frac{w_{ik}^t - 2 \cdot d_k(c_{ik}^t)}{4h \cdot 2^k} \\ &\geq \frac{1}{4h \cdot 2^\ell} \cdot \left( \sum_{k=0}^{\ell} w_{ik}^t - 2 \cdot \sum_{k=0}^{\ell} h \cdot \ln(4e \cdot n) \cdot 2^k \right). \quad (\text{by Lemma 7}) \end{aligned}$$

Hence, again by the choice of  $t$ ,

$$\sum_{k=0}^{\ell} w_{ik}^T = \sum_{k=0}^{\ell} w_{ik}^t \leq 4h \cdot 2^\ell \cdot \ln n + 4h \cdot \ln(4e \cdot n) \cdot 2^\ell < 8h \cdot \ln(4e \cdot n) \cdot 2^\ell. \quad \blacktriangleleft$$

**Bounding Node Degrees.** Recall that whenever a new hyperedge  $S$  containing node  $i$  arrives,  $p_S = \min_{i \in S} p(i)$  is determined. Then, for each node  $i \in S$ , if  $p(i) \leq p_S + h - 1$ , the counter  $w_{i,p(i)}$  is incremented. If  $p(i) \geq p_S + h$ , the degree of  $i$  grows, but  $w_{i,p(i)}$  is not incremented. To estimate the degree of  $i$ , we therefore introduce the counters  $s_{ik}$ , which are incremented in the latter case. That is, for each node  $i \in S$ , we always have  $\sum_{k \geq 0} (w_{ik} + s_{ik}) = \deg(i)$ .

The growth of the variables  $s_{ik}$  is not controlled by the potential, but they grow only for nodes whose degree is very high compared to the current minimum degree. By constructing an appropriate charging argument, we can link their growth to the growth of other variables  $w_{ik}$ .

► **Lemma 9.** Fix a step  $t \leq T$ , a node  $i$ , and a phase  $\ell \geq 0$ . Then,  $s_{i\ell}^t \leq \sum_{j \in [n]} \sum_{r=0}^{\ell-h} w_{jr}^t$ .

**Proof.** We fix node  $i$ , phase  $\ell \geq 0$ , and show the lemma by induction on  $t$ . The inductive basis is trivial, as  $s_{i\ell}^0 = 0 = \sum_{j \in [n]} \sum_{r=0}^{\ell-h} w_{jr}^0$ .

Now suppose that the lemma statement holds for step  $t-1$ . We look at how both sides of the inequality change as we increase the step superscripts from  $t-1$  to  $t$ , and argue that the increase of the right hand side is at least as large as the increase of the left hand side. If  $s_{i\ell}$  does not change, the inductive claim follows trivially. Otherwise,  $s_{i\ell}$  is incremented by 1. This happens only if  $\ell = p(i)$ ,  $i \in S$ , and  $\ell \geq p_S + h$ . By the definition of  $p_S$ , this means that there exists at least one node  $j^* \in S$  such that  $p(j^*) = p_S$ , and the corresponding counter  $w_{j^*,p_S}$  is also incremented. This means that the right hand side of the lemma inequality is incremented by at least  $\sum_{j \in [n]} \sum_{r=0}^{\ell-h} (w_{jr}^t - w_{jr}^{t-1}) \geq w_{j^*,p_S}^t - w_{j^*,p_S}^{t-1} = 1$ , and the inductive claim follows.  $\blacktriangleleft$

► **Lemma 10.** Fix a node  $i$  and a phase  $\ell \geq 0$ . If the event  $\mathcal{T}_E$  occurs, then  $\sum_{k=0}^{\ell} s_{ik}^T \leq 16h \cdot \ln(4e \cdot n) \cdot 2^\ell$ .

## 18:12 Online Disjoint Set Covers: Randomization Is Not Necessary

**Proof.** By Lemma 9,

$$\begin{aligned} s_{ik}^T &\leq \sum_{j \in [n]} \sum_{r=0}^{k-h} w_{jr}^T \leq n \cdot 8h \cdot \ln(4e \cdot n) \cdot 2^{k-h} && \text{(by Lemma 8)} \\ &\leq 8h \cdot \ln(4e \cdot n) \cdot 2^k. && \text{(as } h = \lceil \log n \rceil) \end{aligned}$$

Hence,  $\sum_{k=0}^{\ell} s_{ik}^T < 16h \cdot \ln(4e \cdot n) \cdot 2^{\ell}$ .  $\blacktriangleleft$

Finally, we can show Lemma 2, restated below.

► **Lemma 2.** *Fix a sequence  $E$  such that  $\delta(E) > 24h \cdot \ln(4e \cdot n) \cdot 2^{\ell}$  for some integer  $\ell \geq 0$ . If  $\mathcal{T}_E$  occurs, then each node has completed its phase  $\ell$ .*

**Proof.** Suppose for a contradiction that there exists a node  $i$  for which  $p(i) \leq \ell$  at the end of the input. Then,

$$\delta(E) \leq \deg^T(i) = \sum_{k \geq 0} (w_{ik}^T + s_{ik}^T) = \sum_{k=0}^{\ell} (w_{ik}^T + s_{ik}^T) \leq 24h \cdot \ln(4e \cdot n) \cdot 2^{\ell},$$

where the last inequality follows by Lemma 8 and Lemma 10. This would contradict the assumption of the lemma.  $\blacktriangleleft$

### 4 Controlling the Potential

In this section, we show that  $\{\Phi^t\}_{t \geq 0}$  is a supermartingale with respect to the choices of RAND made in corresponding steps, i.e., we prove Lemma 4.

Throughout this section, we focus on a single step  $t$ , in which RAND processes a hyperedge  $S$ . Recall that RAND first chooses a random integer  $k^*$  uniformly from the set  $\{p_S, p_S + 1, \dots, p_S + h - 1\}$ . Second, conditioned on the first choice, it chooses a random color uniformly from the set  $R_{k^*}$ .

By the definition of our variables, for each node  $i$  and each integer  $k \geq 0$ , it holds that  $w_{ik}^t - w_{ik}^{t-1} \in \{0, 1\}$  and  $c_{ik}^t - c_{ik}^{t-1} \in \{0, 1\}$ .

► **Lemma 11.** *Fix a node  $i \in S$  and let  $p = p^{t-1}(i)$ . If  $p \leq p_S + h - 1$ , then  $c_{ip}^t = c_{ip}^{t-1} + 1$  with probability  $(2^p - c_{ip}^{t-1})/(h \cdot 2^p)$ .*

**Proof.** By the definition of  $p_S$ , we have  $p \geq p_S$ . Combining this with the lemma assumption, we get  $p \in \{p_S, \dots, p_S + h - 1\}$ .

Note that  $c_{ip}^t = c_{ip}^{t-1} + 1$  when node  $i$  gathers a new color from  $R_p$ , and  $c_{ip}^t = c_{ip}^{t-1}$  otherwise. For a node  $i$  to gather a new color from  $R_p$ , first the integer  $k^*$  chosen randomly from the set  $\{p_S, \dots, p_S + h - 1\}$  must be equal to  $p$ , which happens with probability  $1/h$ . Second, conditioned on the former event, a color chosen randomly from the palette  $R_p$  must be different from all  $c_{ip}^{t-1}$  colors from  $R_p$  gathered so far by node  $i$ , which happens with probability  $(|R_p| - c_{ip}^{t-1})/|R_p| = (2^p - c_{ip}^{t-1})/2^p$ . Hence, the probability of gathering a new color is  $(2^p - c_{ip}^{t-1})/(h \cdot 2^p)$ .  $\blacktriangleleft$

We emphasize that the relations involving random variables in the following lemma (e.g., the statements such as  $Z_i^t \leq Z_i^{t-1}$ ) hold for all random choices made by RAND.

► **Lemma 12.** Fix a node  $i$ . Let  $p = p^{t-1}(i)$ . Then, either  $Z_i^t \leq Z_i^{t-1}$  or

$$Z_i^t \leq Z_i^{t-1} + \frac{1}{4h \cdot 2^p} + \begin{cases} -1/(2h \cdot 2^p \cdot \alpha) & \text{with probability } \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

where  $\alpha = (2^p - c_{ip}^{t-1})/(h \cdot 2^p)$ . The probability is computed exclusively with respect to random choices of RAND in step  $t$ .

**Proof.** For brevity, for an integer  $k \geq 0$ , we define

$$\begin{aligned} \Delta w_{ik} &= w_{ik}^t - w_{ik}^{t-1}, \\ \Delta d_k(c_{ik}) &= d_k(c_{ik}^t) - d_k(c_{ik}^{t-1}). \end{aligned}$$

As  $c_{ik}^t \geq c_{ik}^{t-1}$  and  $d_k$  is a non-decreasing function, we have  $\Delta d_k(c_{ik}) \geq 0$ .

Let  $S$  be the hyperedge presented in step  $t$ . By the definition of the variables  $w_{ik}$  (cf. Subsection 2.2), we have

$$\Delta w_{ik} = \begin{cases} 1 & \text{if } i \in S \text{ and } k = p \text{ and } p \leq p_S + h - 1, \\ 0 & \text{otherwise.} \end{cases}$$

Now we consider two cases.

■ It holds that  $i \notin S$  or  $p \geq p_S + h$ . Then,

$$Z_i^t - Z_i^{t-1} = \sum_{k \geq 0} \frac{\Delta w_{ik} - 2 \cdot \Delta d_k(c_{ik})}{4h \cdot 2^k} \leq \sum_{k \geq 0} \frac{\Delta w_{ik}}{4h \cdot 2^k} = 0.$$

■ It holds that  $i \in S$  and  $p \geq p_S + h - 1$ . Then,  $\Delta w_{ip} = 1$  and  $\Delta w_{ik} = 0$  for  $k \neq p$ . Therefore,

$$\begin{aligned} Z_i^t - Z_i^{t-1} &= \sum_{k \geq 0} \frac{\Delta w_{ik} - 2 \cdot \Delta d_k(c_{ik})}{4h \cdot 2^k} \\ &= \frac{\Delta w_{ip} - 2 \cdot \Delta d_p(c_{ip})}{4h \cdot 2^p} + \sum_{k \neq p} \frac{\Delta w_{ik} - 2 \cdot \Delta d_k(c_{ik})}{4h \cdot 2^k} \\ &\leq \frac{1}{4h \cdot 2^p} - \frac{\Delta d_p(c_{ip})}{2h \cdot 2^p}. \end{aligned}$$

To complete the proof, it now suffices to argue that  $\Delta d_p(c_{ip}^t) = 1/\alpha$  with probability  $\alpha$  and 0 with the remaining probability. This follows immediately by Lemma 11: With probability  $\alpha$  we have  $c_{ip}^t = c_{ip}^{t-1} + 1$ , and hence  $\Delta d_p(c_{ip}) = d_p(c_{ip}^t) - d_p(c_{ip}^{t-1}) = h \cdot 2^p / (2^p - c_{ip}^t + 1) = h \cdot 2^p / (2^p - c_{ip}^{t-1}) = 1/\alpha$ . With the remaining probability  $c_{ip}^t = c_{ip}^{t-1}$  and thus  $\Delta d_p(c_{ip}) = 0$ . ◀

For the final lemma, we need the following technical bound (proven in Appendix A). This can be seen as a reverse Jensen's type inequality.

▷ **Claim 13.** Fix  $\varepsilon \in [0, 1]$ ,  $\alpha \in [\varepsilon, 1]$  and a real  $x$ . Let  $X$  be a random variable such that

$$X = \begin{cases} x - \varepsilon/\alpha & \text{with probability } \alpha, \\ x & \text{otherwise.} \end{cases}$$

Then,  $\mathbb{E}[e^{-X}] \leq e^{x-\varepsilon/2}$ .

Finally, we can prove Lemma 4, restated below.

► **Lemma 4.** *Fix a step  $t$  and an outcome of random choices till step  $t - 1$  inclusively. (In particular, this will fix the value of  $\Phi^{t-1}$ .) Then,  $\mathbb{E}[\Phi^t] \leq \Phi^{t-1}$ , where the expectation is taken exclusively over random choices of RAND in step  $t$ .*

**Proof.** Fix a node  $i$ . We will show that

$$\mathbb{E}[\exp(Z_i^t)] \leq \exp(Z_i^{t-1}). \quad (3)$$

The lemma then follows by summing the above inequality over all nodes.

If  $Z_i^t \leq Z_i^{t-1}$ , (3) follows trivially. Otherwise, Lemma 12 implies that

$$Z_i^t \leq Z_i^{t-1} + \frac{1}{4h \cdot 2^p} + \begin{cases} -1/(2h \cdot 2^p \cdot \alpha) & \text{with probability } \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

where  $p = p^{t-1}(i)$  and  $\alpha = (2^p - c_{ip}^{t-1})/(h \cdot 2^p)$ . As the random choices are fixed until step  $t - 1$ , the variables  $Z_i^{t-1}$ ,  $p$  and  $\alpha$  are no longer random variables, but real numbers.

Note that  $c_{ip}^{t-1} \leq q_p - 1 \leq 2^p - 1$  as otherwise phase  $p$  of node  $i$  would have ended in an earlier step. Hence,  $\alpha = (2^p - c_{ip}^{t-1})/(h \cdot 2^p) \geq 1/(2h \cdot 2^p)$ .

Thus,  $x = Z_i^{t-1} + 1/(4h \cdot 2^p)$ ,  $\varepsilon = 1/(2h \cdot 2^p)$ , and  $\alpha$  satisfy the conditions of Claim 13. (In particular,  $\varepsilon \leq \alpha \leq 1$ .) This claim now yields

$$\mathbb{E}[\exp(Z_i^t)] \leq \exp\left(Z_i^{t-1} + \frac{1}{4h \cdot 2^p} - \frac{1}{2} \cdot \frac{1}{2h \cdot 2^p}\right) = \exp(Z_i^{t-1}),$$

which concludes the proof of (3), and thus also the lemma. ◀

## 5 Conclusions

In this paper, we have constructed a deterministic  $O(\log^2 n)$ -competitive algorithm for the disjoint set covers (DSC) problem. Closing the remaining logarithmic gap between the current upper and lower bounds is an interesting open problem that seems to require a new algorithm that goes beyond the phase-based approach.

We have developed new derandomization techniques that extend the existing potential function methods. We hope that these extensions will be useful for derandomizing other online randomized algorithms, and eventually for providing a coherent online derandomization toolbox.

---

## References

- 1 Anna Adamaszek, Artur Czumaj, Matthias Englert, and Harald Räcke. An  $O(\log k)$ -competitive algorithm for generalized caching. *ACM Transactions on Algorithms*, 15(1):6:1–6:18, 2019. doi:10.1145/3280826.
- 2 Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009. doi:10.1137/060661946.
- 3 Noga Alon and Joel H. Spencer. *The Probabilistic Method, Second Edition*. John Wiley, 2000. doi:10.1002/0471722154.
- 4 Piotr Berman, Gruia Călinescu, C. Shah, and Alexander Zelikovsky. Power efficient monitoring management in sensor networks. In *2004 IEEE Wireless Communications and Networking Conference*, pages 2329–2334. IEEE, 2004. doi:10.1109/WCNC.2004.1311452.

- 5 Marcin Bienkowski, Björn Feldkord, and Pawel Schmidt. A nearly optimal deterministic online algorithm for non-metric facility location. In *Proc. 38th Symp. on Theoretical Aspects of Computer Science (STACS)*, LIPIcs, pages 14:1–14:17, 2021. doi:10.4230/LIPIcs.STACS.2021.14.
- 6 Béla Bollobás, David Pritchard, Thomas Rothvoß, and Alex D. Scott. Cover-decomposition and polychromatic numbers. *SIAM Journal on Discrete Mathematics*, 27(1):240–256, 2013. doi:10.1137/110856332.
- 7 Alan Borodin, Nati Linial, and Michael E. Saks. An optimal on-line algorithm for metrical task system. *Journal of the ACM*, 39(4):745–763, 1992. doi:10.1145/146585.146588.
- 8 Sébastien Bubeck, Christian Coester, and Yuval Rabani. The randomized k-server conjecture is false! In *Proc. 55th ACM Symp. on Theory of Computing (STOC)*, pages 581–594. ACM, 2023. doi:10.1145/3564246.3585132.
- 9 Sébastien Bubeck, Michael B. Cohen, James R. Lee, and Yin Tat Lee. Metrical task systems on trees via mirror descent and unfair gluing. In *Proc. 30th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 89–97. SIAM, 2019. doi:10.1137/1.9781611975482.6.
- 10 Niv Buchbinder and Joseph (Seffi) Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009. doi:10.1287/MOOR.1080.0363.
- 11 Adam L. Buchsbaum, Alon Efrat, Shaili Jain, Suresh Venkatasubramanian, and Ke Yi. Restricted strip covering and the sensor cover problem. In *Proc. 18th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1056–1063. SIAM, 2007. URL: <http://dl.acm.org/citation.cfm?id=1283383.1283497>, doi:10.5555/1283383.1283497.
- 12 Mihaela Cardei and Ding-Zhu Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3):333–340, 2005. doi:10.1007/S11276-005-6615-6.
- 13 Mihaela Cardei, My T. Thai, Yingshu Li, and Weili Wu. Energy-efficient target coverage in wireless sensor networks. In *Proc. 2005 IEEE Int. Conf. on Computer Communications (INFOCOM)*, pages 1976–1984. IEEE, 2005. doi:10.1109/INFOCOM.2005.1498475.
- 14 Yuval Emek, Adam Goldbraikh, and Erez Kantor. Online disjoint set cover without prior knowledge. In *Proc. 27th European Symp. on Algorithms (ESA)*, LIPIcs, pages 44:1–44:16, 2019. doi:10.4230/LIPIcs.ESA.2019.44.
- 15 Uriel Feige, Magnús M. Halldórsson, Guy Kortsarz, and Aravind Srinivasan. Approximating the domatic number. *SIAM Journal on Computing*, 32(1):172–195, 2002. doi:10.1137/S0097539700380754.
- 16 Matt Gibson and Kasturi R. Varadarajan. Optimally decomposing coverings with translates of a convex polygon. *Discret. Comput. Geom.*, 46(2):313–333, 2011. doi:10.1007/S00454-011-9353-9.
- 17 Ram P. Gupta. On the chromatic index and the cover index of a multigraph. In *Theory and Applications of Graphs*, pages 204–215. Springer Berlin Heidelberg, 1978.
- 18 Ngoc Mai Le, Seeun William Umboh, and Ningyuan Xie. The power of clairvoyance for multi-level aggregation and set cover with delay. In *Proc. 2023 ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1594–1610. SIAM, 2023. doi:10.1137/1.9781611977554.CH59.
- 19 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, USA, 2nd edition, 2017.
- 20 Ashwin Pananjady, Vivek Kumar Bagaria, and Rahul Vaze. The online disjoint set cover problem and its applications. In *Proc. 2015 IEEE Int. Conf. on Computer Communications (INFOCOM)*, pages 1221–1229. IEEE, 2015. doi:10.1109/INFOCOM.2015.7218497.
- 21 Ashwin Pananjady, Vivek Kumar Bagaria, and Rahul Vaze. Optimally approximating the coverage lifetime of wireless sensor networks. *IEEE/ACM Transactions on Networking*, 25(1):98–111, 2017. doi:10.1109/TNET.2016.2574563.

## 18:16 Online Disjoint Set Covers: Randomization Is Not Necessary

- 22 Prabhakar Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *Journal of Computer and System Sciences*, 37(2):130–143, 1988. doi:10.1016/0022-0000(88)90003-7.
- 23 Neal E. Young. On-line file caching. *Algorithmica*, 33(3):371–383, 2002. doi:10.1007/S00453-001-0124-5.

### A Proofs of Technical Claims

▷ Claim 6. For each  $k \geq 0$  it holds that  $H(2^k) - H(2^k - q_k) \leq \ln(4e \cdot n)$ .

Proof. Assume first that  $2^k < 4n$ . As  $H(m) \leq 1 + \ln m = \ln(e \cdot m)$  for each  $m$ , we get  $H(2^k) - H(2^k - q_k) < H(4n) \leq \ln(4e \cdot n)$  and the lemma follows.

Hence, in the following, we assume that  $2^k \geq 4n$ , obtaining

$$\begin{aligned} 2^k - q_k &= 2^k - \lceil (1 - 1/(2n)) \cdot 2^k \rceil && \text{(by the definition of } q_k) \\ &\geq 2^k - (1 - 1/(2n)) \cdot 2^k - 1 \\ &= 2^k/(2n) - 1 \\ &\geq 2^k/(4n). && \text{(as } 2^k \geq 4n) \end{aligned}$$

As  $q_k \leq 2^k$ , we can use the relation  $H(m) \geq \ln m$  holding for every  $m \geq 0$ , obtaining

$$H(2^k) - H(2^k - q_k) \leq \ln(e \cdot 2^k) - \ln(2^k - q_k) \leq \ln(e \cdot 2^k) - \ln(2^k/(4n)) = \ln(4e \cdot n). \quad \triangleleft$$

▷ Claim 13. Fix  $\varepsilon \in [0, 1]$ ,  $\alpha \in [\varepsilon, 1]$  and a real  $x$ . Let  $X$  be a random variable such that

$$X = \begin{cases} x - \varepsilon/\alpha & \text{with probability } \alpha, \\ x & \text{otherwise.} \end{cases}$$

Then,  $\mathbb{E}[e^X] \leq e^{x - \varepsilon/2}$ .

Proof. Using  $\alpha \geq \varepsilon$ , we obtain

$$(1 + \varepsilon/\alpha) \cdot (1 - \varepsilon/(2\alpha)) = 1 + \varepsilon/(2\alpha) - \varepsilon^2/(2\alpha^2) \geq 1. \quad (4)$$

Next, we use the relation  $e^{-y} \leq (1 + y)^{-1}$  (holding for every  $y > -1$ ) and (4) to argue that

$$\exp(-\varepsilon/\alpha) \leq (1 + \varepsilon/\alpha)^{-1} \leq 1 - \varepsilon/(2\alpha). \quad (5)$$

Finally, this gives

$$\begin{aligned} \mathbb{E}[e^X] &= \alpha \cdot \exp(x - \varepsilon/\alpha) + (1 - \alpha) \cdot \exp(x) \\ &= e^x \cdot (\alpha \cdot \exp(-\varepsilon/\alpha) + 1 - \alpha) \\ &\leq e^x \cdot (\alpha - \varepsilon/2 + 1 - \alpha) && \text{(by (5))} \\ &\leq e^{x - \varepsilon/2}. && \text{(as } 1 - \varepsilon/2 \leq e^{-\varepsilon/2} \text{ for each } \varepsilon) \quad \triangleleft \end{aligned}$$