

Approximating the Interval Constrained Coloring Problem

Ernst Althaus* Stefan Canzar* Andreas Karrenbauer* Khaled Elbassioni*
Julián Mestre*[†]

Abstract

We consider the *interval constrained coloring* problem, which appears in the interpretation of experimental data in biochemistry. Monitoring hydrogen-deuterium exchange rates via mass spectroscopy experiments is a method used in that field to obtain information about protein tertiary structure. The output of these experiments provides data about the exchange rate of residues in overlapping fragments of the protein backbone. These fragments must be re-assembled in order to obtain a global picture of the protein structure. The *interval constrained coloring* problem is the mathematical abstraction of this re-assembly process.

The objective of the interval constrained coloring problem is to assign a color (exchange rate) to a set of integers (protein residues) such that a set of constraints is satisfied. Each constraint is made up of a closed interval (protein fragment) and requirements on the number of elements that belong to each color class (exchange rates observed in the experiments).

We show that the problem is NP-complete for arbitrary number of colors and we provide algorithms that given a feasible instance find a coloring that satisfies all the coloring requirements within ± 1 of the prescribed value. In light of our first result, this is essentially the best one can hope for. Our approach is based on polyhedral theory techniques and randomized rounding. Furthermore, we develop a quasi-polynomial-time approximation scheme for a variant of our problem where we are asked to find a coloring satisfying as many as fragments as possible.

1 Introduction

Our motivation for the *interval constrained coloring* problem comes from an application in biochemistry. The problem has been introduced recently in [1]. To be self-contained, we restrict ourselves to a very brief and informal description in this paper and refer the interested reader to the publication mentioned above.

A challenging and important problem in biochemistry is to determine the tertiary structure of a protein, i.e. the spatial arrangement, which is indispensable for its function. There are various approaches each with advantages and drawbacks. One method for this task is the so-called *hydrogen-deuterium exchange*, abbreviated by HDX. This is a chemical reaction where a hydrogen atom of the protein is replaced by a deuterium atom, or vice versa. To this end, the protein solution is diluted by D_2O . Intuitively, the exchange process happens at a higher rate at amino acids, or residues, that are more exposed to the solvent. Put differently, the exchange rates for residues at the outside of the complex are higher than inside. Note that though deuterium is heavier than hydrogen, they are almost identically from a chemical point of view. Hence, the exchange rate may be monitored by mass spectroscopy while the tertiary structure remains unaffected by the process. However, this method does not deliver that fine grained information such that the exchange rate for each residue can be determined directly. Rather, we get bulk information for fragments of the protein. For example, we get the number

*Max-Planck-Institute für Informatik. Saarbrücken, Germany.

[†]Supported by an Alexander von Humboldt fellowship.

of slow, medium, and fast residues for each of several overlapping fragments covering the whole protein. That is, the experimental data only tells us how many residues of a fragment react at low, medium, and high exchange rate, respectively. Moreover, we know the exact location and size of each fragment in the protein. It remains to find a valid assignment of all residues to exchange rates that matches the experimentally found bulk information. If the solution is not unique, we want to enumerate all feasible of them or a representative subset thereof as a basis for further chemical considerations.

The problem can be rephrased in mathematical terms as follows. We are given a protein of n residues and a set of fragments, which correspond to intervals of $[n]$. The fragments cover the whole protein and may overlap. Furthermore, there are k possible exchange rates to which we refer as colors in the following. In the *interval constrained coloring* problem, the goal is to produce a coloring of the set $[n]$ using k colors such that a given set of requirements is satisfied. Each requirement is made up of a closed interval $I \subseteq [n]$ and a complete specification of how many elements in I should be colored with each color class.

More formally, let \mathcal{I} be a set of intervals defined on the set $V = [n]$, let $[k]$ be a set of color classes, and let $r : \mathcal{I} \times [k] \rightarrow \mathbb{Z}^+$ be a requirement function such that $\sum_{c \in [k]} r(I, c) = |I|$ for all $I \in \mathcal{I}$. A coloring $\chi : V \rightarrow [k]$ is said to be *feasible* if for every $I \in \mathcal{I}$ we have

$$|\{i \in I \mid \chi(i) = c\}| = r(I, c) \text{ for all } c \in [k] \quad (1)$$

Given this information, we would like to determine whether or not a feasible coloring exists, and if so, to produce one.

The problem is captured by the integer program given below. The binary variable $x_{i,c}$ indicates whether i is colored c or not. Constraint (2) enforces that each residue gets exactly one color and constraint (3) enforces that every requirement is satisfied.

$$\sum_{c \in [k]} x_{i,c} = 1 \quad \forall i \in [n] \quad (2)$$

$$\sum_{i \in I} x_{i,c} = r(I, c) \quad \forall I \in \mathcal{I}, c \in [k] \quad (3)$$

$$x_{i,c} \in \{0, 1\} \quad \forall i \in [n], c \in [k] \quad (4)$$

Let \mathcal{P} be the polytope obtained by relaxing the integrality constraint (4) in the above integral problem. That is \mathcal{P} is the set of values of x obeying (2), (3) and $0 \leq x_{i,c} \leq 1$ for all i and c .

1.1 Previous and Related Work

The polyhedral description has already been introduced in [1] and has served there as a basis to attack the problem by integer programming methods and tools, which perform well in practice. Moreover, the authors established the polynomial-time solvability of the two-color case by the integrality of the polytope \mathcal{P} and provided also a combinatorial algorithm for this case. However, the complexity of the general problem has been left open.

A closely related problem is *broadcast scheduling*, where a server must decide which data item to broadcast at each time step in order to satisfy client requests. The literature in broadcast scheduling is vast and many variations of the problem have been studied (see [2, 4] and references therein). In the variant we are concerned with here, a client request is specified by a time window I and a data type A . The request is satisfied if A is broadcast *at least once* in I . The similarities between the two problems should be clear with time steps, time windows and data types in broadcast scheduling playing the respective roles of positions, intervals and colors in interval constrained coloring. There are, however, important differences. First, whereas in broadcast scheduling it does not hurt to broadcast an item more times than the prescribed number, in our problem it does. Second, an interval is satisfied only if *all* the requirements for that interval are satisfied *exactly*, which, undoubtedly, makes our problem significantly harder.

1.2 Contributions of this Paper

As mentioned above, the complexity status for the *interval constrained coloring* problem has been open. In Section 4 we partly settle this by showing that deciding whether a feasible coloring exists is NP-complete when k is part of the input.

Although the polytope \mathcal{P} is integral for $k = 2$, it need to be for $k > 3$. Nevertheless, we can check in polynomial time whether $\mathcal{P} = \emptyset$. If that is the case then we know that there is no feasible coloring. Otherwise we can find a feasible fractional solution. In Section 2 we will show how to round this fractional solution to produce a coloring where *all* the requirements are satisfied within a mere additive error of one.

In practice, the data emanating from the experiments is noisy, which normally causes the instance to be infeasible and in some case even forces \mathcal{P} to be empty. To deal with this problem in Section 3 we study a variant of the problem in which we want to maximize the number of requirements that are satisfied. Another way to deal with noisy data is to model the noise in the linear programming relaxation to get a new set of requirements on which to run the algorithm from Section 2. The latter approach was explored by Althaus *et al.* [1]; the reader is referred to their paper for details.

2 A ± 1 guarantee

Let x be a fractional solution in \mathcal{P} . We use the scheme of Gandhi *et al.* [4] to round x to an integral solution \hat{x} with the following properties:

Theorem 1. *Given a fractional solution $x \in \mathcal{P}$ we can construct in polynomial time an integral solution \hat{x} with the following properties*

(P1) *For every $i \in [n]$ there exists $c \in [k]$ such that $\hat{x}_c = 1$ and $\hat{x}_d = 0$ for all $d \neq c$.*

(P2) *For every $I \in \mathcal{I}$ and $c \in [k]$ we have $|\sum_{i \in I} \hat{x}_{i,c} - r(I, c)| \leq 1$.*

(P3) *Every $I \in \mathcal{I}$ is satisfied with probability greater or equal than $\gamma_k = \frac{k(k+1-H_{k-1})}{(k+1)!}$.*

In other words, each position gets exactly one color (P1), every coloring requirement is off by at most one from the prescribed number (P2), and all the requirements for a given interval I are satisfied *exactly* ($\sum_{i \in I} \hat{x}_{i,c} = r(I, c)$ for all $c \in [k]$) with probability at least γ_k . An interesting corollary of this theorem is that if \mathcal{P} is non-empty then there exists always a coloring satisfying at least $\gamma_k |\mathcal{I}|$ intervals, and such coloring can be found in polynomial time.

The high level idea is to simplify the polytope \mathcal{P} into another integral polytope with basic solutions satisfying (P1) and (P2). Then we show how to select a basic solution satisfying (P3). This is done by defining a set of *blocks* and then setting up an assignment problem instance between $[n]$ and the set of blocks, whose polytope is integral.

For each color class $c \in [k]$ we choose a real number $\alpha_c \in [0, 1]$, to be specified shortly. Let us define blocks $B_1^c, B_2^c, \dots, B_{b_c}^c$: For color c and $j = 2, \dots, b_c - 1$

$$B_j^c = \left[\min\{t \mid \sum_{i \leq t} x_{i,c} > j - 2 + \alpha_c\}, \min\{t \mid \sum_{i \leq t} x_{i,c} \geq j - 1 + \alpha_c\} \right]. \quad (5)$$

The first and last blocks, B_1^c and $B_{b_c}^c$, are defined similarly, but starting at 1 and ending at n respectively.

For each $i \in B_j^c$ we define a variable $y_{i,(c,j)}$. If i belongs to a single block B_j^c of color c then we set $y_{i,(c,j)} = x_{i,c}$. Otherwise, i belongs to two adjacent blocks B_{j+1}^c and B_j^c , in which case we set $y_{i,(c,j+1)} = \sum_{t \leq i} x_{t,c} - (j - 1 + \alpha_c)$ and $y_{i,(c,j)} = x_{i,c} - y_{i,(c,j+1)}$. See Figure 1 for an example of how the blocks and the solution y are constructed. Another, equivalent, way to define y is to ask that $x_{i,c} = \sum_j y_{i,(c,j)}$, $\sum_{i \in B_1^c} y_{i,(c,1)} = \alpha_c$ and $\sum_{i \in B_j^c} y_{i,(c,j)} = 1$ for every $1 < j < b_c$.

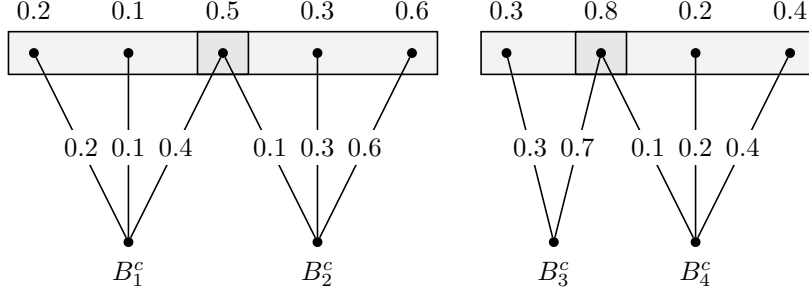


Figure 1: How the blocks B_j^c are constructed. The $x_{i,c}$ values appear on top and the $y_{i,(c,j)}$ values appear on the edges. Note that a block can only overlap with its predecessor or successor. In this case $\alpha_c = 0.7$.

Thus y defines a feasible fractional assignment between $[n]$ and the set of blocks. Let \mathcal{Q} be the polytope of this assignment problem, namely,

$$\sum_{B_j^c \ni i} y_{i,(c,j)} = 1 \quad \forall i \in [n] \quad (6)$$

$$\sum_{i \in B_j^c} y_{i,(c,j)} = 1 \quad \forall c \in [k] \text{ and } 1 < j < b_c \quad (7)$$

$$\sum_{i \in B_j^c} y_{i,(c,j)} \leq 1 \quad \forall c \in [k] \text{ and } j \in \{1, b_c\} \quad (8)$$

$$y_{i,(c,t)} \geq 0 \quad \forall i \in [n], c \in [k], t \in [b_c] \quad (9)$$

Because \mathcal{Q} is integral, any fractional solution $y \in \mathcal{Q}$ can be turned into an integral solution $\hat{y} \in \mathcal{Q}$; this can even be done in polynomial time. Notice that an integral solution \hat{y} to \mathcal{Q} induces an integral solution \hat{x} by setting $\hat{x}_{i,c} = 1$ if and only if $y_{i,(c,j)} = 1$. Constraint (6) implies that \hat{x} satisfies (P1). Furthermore, \hat{x} also satisfies (P2).

Lemma 1. *Let \hat{y} be an integral solution for \mathcal{Q} and let \hat{x} be the coloring induced by \hat{y} . Then $|\sum_{i \in I} \hat{x}_{i,c} - r(I, c)| \leq 1$ for all $I \in \mathcal{I}$ and $c \in [k]$.*

Proof. Since $\sum_{i \in I} x_{i,c} = r(I, c)$, the number of blocks of color c that intersect I is either $r(I, c)$ or $r(I, c) + 1$. Furthermore, at least $r(I, c) - 1$ of these blocks lie entirely within I and at most two blocks intersecting I partially. Due to constraint (7), each internal block will force a different position in I to be colored c . On the other hand, the fringe blocks, if any, can force at most two additional positions in I to be colored c . Hence, the lemma follows. \square

It only remains to prove that \hat{x} obeys (P3). To do so, we need to introduce some randomization in our construction. First, we will choose the offset α_c of each color $c \in [k]$ independently and uniformly at random. Second, instead of choosing any extreme point of \mathcal{Q} , we choose one using a randomized rounding procedure.

Gandhi *et al.* [4] showed that any fractional solution $y \in \mathcal{Q}$ can be rounded to an integral solution $\hat{y} \in \mathcal{Q}$ such that the probability that $\hat{y}_{i,(c,j)} = 1$ is exactly $y_{i,(c,j)}$. It is important to note that these events are not independent of each other.

Lemma 2. *Let \hat{y} be the solution output by the randomized rounding procedure and \hat{x} the coloring induced by it. For any interval $I \in \mathcal{I}$, the probability that $\sum_{i \in I} \hat{x}_{i,c} = r(I, c)$ for all $c \in [k]$ is at least $\frac{k(k+1-H_{k-1})}{(k+1)!}$.*

Proof. Let I be an arbitrary, but fixed, interval throughout the proof and for time being let us concentrate on a fixed, but arbitrary, color $c \in [k]$. Let f and l be the indices of the first and

last blocks of color class c that intersect I and define $\beta_c = \sum_{i \in I \cap B_f^c} x_{i,(c,f)}$, or, equivalently, $\sum_{i \in I \cap B_f^c} x_{i,(c,l)} = 1 - \beta_c$.

Intuitively, the probability that $\sum_{i \in I} \hat{x}_{i,c} = r(I, c)$ should be greater when the blocks of c are aligned with I (when β_c is close to 0 or 1) and it should be low when they are not (when β_c is around 0.5). By choosing α_c uniformly at random, β_c also becomes a random variable uniformly distributed in $[0, 1]$. Thus, we have a decent chance of getting a “good value” of β_c .

Let us formalize and make more precise the above idea. Denote with ξ_f and ξ_l the events $\sum_{i \in I \cap B_f^c} \hat{y}_{i,(c,f)} = 1$ and $\sum_{i \in I \cap B_f^c} \hat{y}_{i,(c,l)} = 1$ respectively. Let $\beta = (\beta_1, \dots, \beta_k)$ be the vector offset for the color classes. For brevity’s sake we denote $\Pr[\xi \mid \beta]$ with $\Pr_\beta[\xi]$.

$$\begin{aligned} \Pr_\beta \left[\sum_{i \in I} \hat{x}_{i,c} \neq r(I, c) \right] &= \Pr_\beta \left[\xi_f \xi_l \vee \overline{\xi_f} \overline{\xi_l} \right] \\ &= \Pr_\beta \left[\xi_f \xi_l \right] + \Pr_\beta \left[\overline{\xi_f} \overline{\xi_l} \right] \\ &\leq \min \{ \Pr_\beta \left[\xi_f \right], \Pr_\beta \left[\xi_l \right] \} + \min \{ \Pr_\beta \left[\overline{\xi_f} \right], \Pr_\beta \left[\overline{\xi_l} \right] \} \end{aligned}$$

Since $\Pr_\beta \left[\xi_f \right] = \beta_c$ and $\Pr_\beta \left[\xi_l \right] = 1 - \beta_c$, it follows that

$$\Pr_\beta \left[\sum_{i \in I} \hat{x}_{i,c} \neq r(I, c) \right] \leq 2 \min \{ \beta_c, 1 - \beta_c \} \quad (10)$$

As a warm-up we first show that the probability that all requirements for I are fulfilled is at least $\frac{1}{(k+1)!}$. Recall that the vector β is distributed uniformly over the domain $D = [0, 1]^k$. Conditioning on β and averaging over D gives the desired result.

$$\begin{aligned} \Pr \left[\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c) \right] &= \int_D \Pr_\beta \left[\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c) \right] d\beta_1 \cdots d\beta_k \\ &\geq \int_D 1 - \sum_{c \in [k]} \Pr_\beta \left[\sum_{i \in I} \hat{x}_{i,c} \neq r(I, c) \right] d\beta_1 \cdots d\beta_k \\ &\geq \int_D \max \left\{ 0, 1 - 2 \sum_{c \in [k]} \min \{ \beta_c, 1 - \beta_c \} \right\} d\beta_1 \cdots d\beta_k \\ &= 2 \int_D \max \left\{ 0, \frac{1}{2} - \sum_{c \in [k]} \min \{ \beta_c, 1 - \beta_c \} \right\} d\beta_1 \cdots d\beta_k \end{aligned}$$

The second inequality follows from the union bound and the third from (10). A moment’s thought reveals that the function inside the integral is symmetrical in the 2^k orthants around the point $(\frac{1}{2}, \dots, \frac{1}{2}) \in D$. Therefore, setting $D' = [0, \frac{1}{2}]^k$ we get

$$\Pr \left[\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c) \right] \geq 2^{k+1} \int_{D'} \max \left\{ 0, \frac{1}{2} - \sum_{c \in [k]} \beta_c \right\} d\beta_1 \cdots d\beta_k$$

The last integral can be interpreted as the volume of a $(k+1)$ -dimensional simplex.

$$\begin{aligned} \Pr \left[\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c) \right] &\geq 2^{k+1} \text{Vol} \left(y \in R_+^{k+1} \mid \sum_{i \in [k+1]} \lambda_i \leq \frac{1}{2} \right) \\ &= 2^{k+1} \frac{\left(\frac{1}{2}\right)^{k+1}}{(k+1)!} \\ &= \frac{1}{(k+1)!} \end{aligned}$$

In order to get the stronger bound in the statement of the lemma we need two more ideas. First, we claim that we only need to condition on fulfilling $k-1$ requirements: Because $\sum_{c \in [k]} r(I, c) = |I|$, once we get $k-1$ colors right, the k th requirement must be satisfied as well. Second, since we can condition on any $k-1$ colors, we had better condition on the ones with smallest offset, that is, those that are close to 0 or 1.

$$\begin{aligned}
\Pr [\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c)] &= \int_D \Pr_\beta [\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c)] d\beta_1 \cdots d\beta_k \\
&\geq \int_D \max_{d \in [k]} \left\{ 1 - \sum_{c \neq d} \Pr_\beta [\sum_{i \in I} \hat{x}_{i,c} \neq r(I, c)] \right\} d\beta_1 \cdots d\beta_k \\
&\geq \int_D \max_{d \in [k]} \left\{ \max \left\{ 0, 1 - 2 \sum_{c \neq d} \min \{ \beta_c, 1 - \beta_c \} \right\} \right\} d\beta_1 \cdots d\beta_k \\
&= 2^k \int_{D'} \max_{d \in [k]} \left\{ \max \left\{ 0, 1 - 2 \sum_{c \neq d} \beta_c \right\} \right\} d\beta_1 \cdots d\beta_k \\
&= 2^{k+1} \int_{D'} \max \left\{ 0, \frac{1}{2} - \sum_{c \in [k]} \beta_c + \max_{d \in [k]} \beta_d \right\} d\beta_1 \cdots d\beta_k
\end{aligned}$$

The last integral can be simplified by assuming that the maximum β_d is attained by the last variable. Of course, the maximum can be any of the k variables, thus these two quantities are related by a factor of k .

$$\Pr [\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c)] \geq k 2^{k+1} \int_0^{\frac{1}{2}} \left[\int_{[0, z]^{k-1}} \max \left\{ 0, \frac{1}{2} - \sum_{c \in [k-1]} \beta_c \right\} d\beta_1 \cdots d\beta_{k-1} \right] dz$$

Let $T(z)$ denote $\text{Vol} \left(\lambda \in R_+^k \mid \sum_{i=1}^k \lambda_i \leq \frac{1}{2} \text{ and } \lambda_1, \dots, \lambda_{k-1} \leq z \right)$. Then we can rewrite the above integral as

$$\Pr [\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c)] \geq k 2^{k+1} \int_0^{\frac{1}{2}} T(z) dz \tag{11}$$

The volume computed by $T(z)$ is not a simplex, but it can be reduced to a summation involving only the volume of simplices using the principle of inclusion/exclusion.

Let $V(\rho)$ denote the volume $\text{Vol} \left(\lambda \in R_+^k \mid \sum_{i=1}^k \lambda_i \leq \rho \right)$ and recall that $V(\rho) = \frac{\rho^k}{k!}$. Consider what happens when $z \in [\frac{1}{4}, \frac{1}{2}]$; clearly $T(z) < V(\frac{1}{2})$ since $V(\frac{1}{2})$ includes points $\lambda \in R_+^k$ such that $\lambda_i > z$ for exactly one coordinate $i \in [k-1]$ (since $z \geq \frac{1}{4}$). Notice that

$$\text{Vol} \left(\lambda \in R_+^k \mid \sum_{i=1}^k \lambda_i \leq \frac{1}{2} \text{ and } \lambda_i > z \right) = V\left(\frac{1}{2} - z\right)$$

Thus $T(z) = V(\frac{1}{2}) - (k-1)V(\frac{1}{2} - z)$ for $z \in [\frac{1}{4}, \frac{1}{2}]$, but $T(z) > V(\frac{1}{2}) - (k-1)V(\frac{1}{2} - z)$ for $z \in [0, \frac{1}{4}]$ since the volume of points y such the constraint $\lambda_i \leq z$ is violated for two coordinates is subtracted twice. To avoid cumbersome notation, assume $V(\rho) = 0$ if $\rho \leq 0$. A simple inclusion/exclusion argument yields

$$T(z) = \sum_{i=0}^{k-1} \binom{k-1}{i} (-1)^i V\left(\frac{1}{2} - iz\right) \tag{12}$$

Plugging (12) into (11) we get

$$\begin{aligned}
\Pr [\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c)] &\geq 2^{k+1} k \left(\int_0^{\frac{1}{2}} V\left(\frac{1}{2}\right) dz + \sum_{i=1}^{k-1} \binom{k-1}{i} (-1)^i \int_0^{\frac{1}{2^i}} V\left(\frac{1}{2} - iz\right) dz \right) \\
&= 2^{k+1} k \left(\int_0^{\frac{1}{2}} \frac{\left(\frac{1}{2}\right)^k}{k!} dz + \sum_{i=1}^{k-1} \binom{k-1}{i} (-1)^i \int_0^{\frac{1}{2^i}} \frac{\left(\frac{1}{2} - iz\right)^k}{k!} dz \right) \\
&= 2^{k+1} k \left(\frac{1}{k! 2^k} z \Big|_0^{\frac{1}{2}} + \sum_{i=1}^{k-1} \binom{k-1}{i} (-1)^i \frac{\left(\frac{1}{2} - iz\right)^{k+1}}{(k+1)! (-i)} \Big|_0^{\frac{1}{2^i}} \right) \\
&= 2^{k+1} k \left(\frac{1}{k! 2^{k+1}} + \sum_{i=1}^{k-1} \binom{k-1}{i} \frac{(-1)^i}{(k+1)! 2^{k+1} i} \right) \\
&= \frac{k}{(k+1)!} \left(k+1 + \sum_{i=1}^{k-1} \binom{k-1}{i} \frac{(-1)^i}{i} \right)
\end{aligned}$$

Using induction on k , it is straightforward to show that the sum in the last line adds up exactly to $-H_{k-1}$. Therefore, we get the desired bound

$$\Pr [\forall c : \sum_{i \in I} \hat{x}_{i,c} = r(I, c)] \geq \frac{k(k+1 - H_{k-1})}{(k+1)!}. \quad \square$$

Remark: In our application domain the goal usually is not to find a single solution, but to generate a number of candidate solutions and let the user choose the one that he finds most interesting or relevant for the specific application. Our framework is amenable to this task since there are very efficient algorithms to enumerate all the integral solutions of \mathcal{Q} [6].

3 Maximum Coloring

In this section we study a variant of the interval constrained coloring to deal with instances that do not admit a feasible coloring. For these instances we consider the problem of finding a coloring that maximizes the number of intervals satisfying (1). More generally, we assume a non-negative weight $w(I)$, associated with each interval $I \in \mathcal{I}$, and seek a subset $\mathcal{I}' \subseteq \mathcal{I}$, maximizing $w(\mathcal{I}') \stackrel{\text{def}}{=} \sum_{I \in \mathcal{I}'} w(I)$, such that there exists a coloring of V satisfying (1) for each $I \in \mathcal{I}'$. We call this problem MAXCOLORING. Let $\text{OPT} \subseteq \mathcal{I}$ be a subset achieving this maximum. For $\alpha \in (0, 1]$ and $\beta \geq 1$, an (α, β) -approximation of the problem is given by a pair (χ, \mathcal{I}') of a subset $\mathcal{I}' \subseteq \mathcal{I}$, and a coloring $\chi : V \mapsto [k]$, such that $\sum_{I \in \mathcal{I}'} w(I) \geq \alpha \cdot w(\text{OPT})$, and $\frac{1}{\beta} r(I, c) \leq N_\chi(I, c) \leq \beta r(I, c)$, where $N_\chi(I, c)$ is the number of positions in I colored c by χ .

Theorem 2. *Consider an instance (V, \mathcal{I}) of MAXCOLORING with $|V| = n$ and $|\mathcal{I}| = m$. Then we can find a $(1, 1 + \epsilon)$ -approximation in quasi-polynomial time $n^{O(\frac{k^2}{\epsilon} \log n \log m)}$, for any $\epsilon > 0$.*

Note that the above bound is quasi-polynomial for $k = \text{polylog}(n, m)$. To prove Theorem 2 we use a similar technique as in [3]. Our approach can be divided into two parts: (i) Reducing the search space, and (ii) developing a dynamic program. We explain these two steps in more details in the next subsections.

3.1 Reducing the search space

Let $\epsilon > 0$ be a given constant. For a vertex $u \in V$ and a set of intervals \mathcal{I} on V , denote respectively by $\mathcal{I}_L(u)$, $\mathcal{I}_R(u)$, and $\mathcal{I}[u]$ the subsets of intervals of \mathcal{I} that lie to the left of u , lie to the right of u , and span u , that is

$$\begin{aligned}
\mathcal{I}_L(u) &= \{[s, t] \in \mathcal{I} : t \leq u - 1\}, \\
\mathcal{I}_R(u) &= \{[s, t] \in \mathcal{I} : s \geq u + 1\}, \\
\mathcal{I}[u] &= \{[s, t] \in \mathcal{I} : s \leq u \leq t\}.
\end{aligned}$$

Denote by $V_L(u)$ and $V_R(u)$ the sets of vertices that lie to the left and right of $u \in V$, respectively: $V_L(u) = \{i \in V : i < u\}$ and $V_R(u) = \{i \in V : i \geq u\}$.

Definition 1. (Assignments) Let $V = \{p, p + 1, \dots, q\}$. An assignment on V is a pair $A = (\mathcal{I}_A, r_A)$ of intervals \mathcal{I}_A on V and a function $r_A : \mathcal{I}_A \times [k] \mapsto \{0, 1, \dots, |V|\}$ such that $r_A(I) \leq r_A(I')$ for all $I, I' \in \mathcal{I}_A$, with $I \subseteq I'$. A is called a left-assignment (respectively, right-assignment) if all intervals in \mathcal{I}_A start at p (respectively, end at q).

In the following, we fix $h = \lceil k \log n / \log(1 + \epsilon) \rceil + 1$.

Definition 2. (ϵ -Partial assignments) Let $u^* \in V$ be a given vertex of $V = \{p, p + 1, \dots, q\}$. A set of $h_1 + h_2 + 4$ intervals $\mathcal{I}_P = \mathcal{I}_{P_l} \cup \mathcal{I}_{P_r}$, $\mathcal{I}_{P_l} = \{I_0, I_1, \dots, I_{h_1}, I_{h_1+1}\}$ and $\mathcal{I}_{P_r} = \{I'_0, I'_1, \dots, I'_{h_2}, I'_{h_2+1}\}$, and an $r_P : \mathcal{I}_P \times [k] \mapsto \{0, 1, \dots, |V|\}$, such that

(R1) all intervals start or end at u^* : $I_j = [u_j, u^*]$ for $j \in \{0, 1, \dots, h_1\}$, $I_{h_1+1} = [p, u^*]$, $I'_j = [u^*, u'_j]$ for $j \in \{0, 1, \dots, h_2\}$, and $I'_{h_2+1} = [u^*, q]$, where $u_{h_1} < u_{h_1-1} < \dots < u_1 < u_0 < u^* < u'_0 < u'_1 < u'_2 < \dots < u'_{h_2}$ (and thus $I_0 \subset I_1 \subset I_2 \subset \dots \subset I_{h_1}$ and $I'_0 \subset I'_1 \subset I'_2 \subset \dots \subset I'_{h_2}$.)

(R2) $r_P(I, c) \leq r_P(I', c)$ for every $I, I' \in \mathcal{I}_P$, with $I \subseteq I'$, and every $c \in [k]$,

(R3) $\sum_{c \in [k]} r_P(I, c) = |I|$ for every $I \in \mathcal{I}_P$,

(R4) for every $I \in \mathcal{I}_P$, there exist $c \in [k]$ and $i \in \mathbb{Z}_+$ such that $r_P(I, c) = \lceil (1 + \epsilon)^i \rceil$, and

(R5) for every $c \in [k]$ and $i \in \mathbb{Z}_+$ with $i \leq \lfloor (\log r_P(I_{h_1+1}, c) / \log(1 + \epsilon)) \rfloor$, there exists $I \in \mathcal{I}_{P_l}$ such that $r_P(I, c) = \lceil (1 + \epsilon)^i \rceil$; similarly, for every $c \in [k]$ and $i \in \mathbb{Z}_+$ with $i \leq \lfloor (\log r_P(I'_{h_2+1}, c) / \log(1 + \epsilon)) \rfloor$, there exists $I' \in \mathcal{I}_{P_r}$ such that $r_P(I', c) = \lceil (1 + \epsilon)^i \rceil$.

will be called an ϵ -partial assignment w.r.t. u^* , denoted by $P = (u^*, \mathcal{I}_P, r_P)$.

The total number $\mu(n)$ of possible ϵ -partial assignments with respect to a given vertex $u^* \in V$ with $|V| = n$ can be bounded as follows: There are at most $n^{h_1+h_2+2}$ possible choices for the points $u_0, u_1, \dots, u_{h_1}, u'_0, u'_1, \dots, u'_{h_2}$. For each interval I , the number of non-negative integer solutions $r_P(I, c)$, $c \in [k]$, satisfying (R3) is $\binom{|I|+k-1}{k-1} \leq [|I|(\ln k + 1) / (k - 1)]^{k-1}$. Letting $i_0 = |I_0|$, $i_j = |I_j \setminus I_{j-1}|$, for $j \in \{0, 1, \dots, h_1\}$, $i'_0 = |I'_0|$, and $i'_j = |I'_j \setminus I'_{j-1}|$, for $j \in \{0, 1, \dots, h_2\}$, we observe by (R2) and (R3) that

$$\begin{aligned}
\mu(n) &\leq n^{h_1+h_2+2} \prod_{j=0}^{h_1+1} \binom{i_j + k - 1}{k - 1} \prod_{j=0}^{h_2+1} \binom{i'_j + k - 1}{k - 1} \\
&\leq n^{h_1+h_2+2} \left(\frac{\ln k + 1}{k - 1} \right)^{(k-1)(h_1+h_2+4)} \left(\prod_{j=0}^{h_1+1} i_j \prod_{j=0}^{h_2+1} i'_j \right)^{k-1} \\
&\leq n^{h_1+h_2+2} \left(\frac{\ln k + 1}{k - 1} \right)^{(k-1)(h_1+h_2+4)} \left(\frac{\sum_{j=0}^{h_1+1} i_j + \sum_{j=0}^{h_2+1} i'_j}{h_1 + h_2 + 4} \right)^{(k-1)(h_1+h_2+4)} \\
&= n^{h_1+h_2+2} \left(\frac{\ln k + 1}{k - 1} \cdot \frac{n}{h_1 + h_2 + 4} \right)^{(k-1)(h_1+h_2+4)} \\
&\leq \left(\frac{\ln k + 1}{k - 1} \cdot n \right)^{2k^2 \frac{\log n}{\log(1+\epsilon)} + 6k}, \tag{13}
\end{aligned}$$

which is $n^{\text{polylog}(n)}$ for every fixed $\epsilon > 0$ and $k = \text{polylog}(n)$.

Let $\chi : V \mapsto [k]$ be a coloring of V and $u^* \in V$ be an arbitrary vertex. We say that an assignment $A = (\mathcal{I}_A, r_A)$ is *consistent* with χ if $N_\chi(I, c) = r_A(I, c)$ for all $c \in [k]$ and $I \in \mathcal{I}_A$. Two assignments P_1 and P_2 are said to be consistent if there exists a coloring χ with which both are consistent.

Lemma 3. *Let χ be a coloring of V and $u^* \in V$ be an arbitrary vertex. Then there exists an ϵ -partial assignment P on V w.r.t. u^* , that is consistent with χ .*

Proof. Assume that $V = \{p, p+1, \dots, q\}$. Clearly, for every $c \in [k]$, the function $N_\chi([u^*, u], c)$ is monotonically increasing on $u \geq u^*$ with a smallest positive increment of 1. This allows us to define P as follows. Let $u'_0 = u^*$. For $j = 1, 2, \dots, h_2$ let $u'_j = \min\{u > u'_{j-1} : N_\chi([u^*, u], c) = \lceil (1 + \epsilon)^j \rceil \text{ for some } i \in \mathbb{Z}_+ \text{ and } c \in [k]\}$, and set $I'_j = [u^*, u'_j]$ and $I_{h_2+1} = [u^*, q]$. The highest index j for which this iteration can be done will be the value of h_2 . Similarly, we define h_1 and the intervals I_j for $j = 0, 1, \dots, h_1 + 1$. Since $N_\chi(I, c) \leq n$ for all intervals $I \subseteq V$ and colors $c \in [k]$, we get $h_1, h_2 \leq k \log n / \log(1 + \epsilon)$. Finally, we define $r_P(I, c) = N_\chi(I, c)$ for all $c \in [k]$ and $I \in \mathcal{I}_P$. \square

3.2 The dynamic Program

The algorithm is shown in Figure 2, and is parameterized with two assignments P_L and P_R , both initially empty. It is based on a divide-and-conquer approach where a point u^* in the middle of V is picked and all intervals containing u^* are evaluated to see if they should be taken into the solution. To do this evaluation conservatively, the procedure iterates over all ϵ -partial assignments P , consistent with P_L and P_R , w.r.t. to the middle vertex u^* , then recurses on the subsets of intervals to the left and right of u^* .

Lemma 4. *Let $\omega = (n, \mathcal{I}, k, r)$ be an instance of MAXCOLORING. If the set of intervals \mathcal{I} can be partitioned into two sets \mathcal{I}_1 and \mathcal{I}_2 , such that for $x \in \{1, 2\}$ it holds*

- (a) $I_i \cap I_j = \emptyset, \forall I_i, I_j \in \mathcal{I}_x$, i.e. intervals are disjoint
- (b) $\bigcup_{I_j \in \mathcal{I}_x} I_j = [s, t]$, i.e. the union of intervals is an interval again

then the feasibility problem for ω can be solved in time $\mathcal{O}(n^k |\mathcal{I}|)$.

Proof. We are given an instance $\omega = (n, \mathcal{I}, k, r)$, $r : \mathcal{I} \times [k] \mapsto \{0, 1, \dots, n\}$, of MAXCOLORING, which allows the partitioning of the interval set \mathcal{I} into two subsets \mathcal{I}_1 and \mathcal{I}_2 , satisfying conditions (a) and (b) in Lemma 4. We represent sets $\mathcal{I}_1, \mathcal{I}_2$ as sequences $([a_1, b_1], [a_2, b_2], \dots, [a_l, b_l])$ and $([a'_1, b'_1], [a'_2, b'_2], \dots, [a'_m, b'_m])$, respectively, where $a_i = b_{i-1} + 1$, for $2 \leq i \leq l$, and similarly $a'_i = b'_{i-1} + 1$, for $2 \leq i \leq m$.

Without loss of generality assume $a_1 \leq a'_1$. Let t be the smallest integer for which $b_t \geq a'_1$. Then instance ω is feasible if and only if

- $\sum_{j=1}^k r([a_i, b_i], j) = b_i - a_i + 1$, for all $1 \leq i \leq t - 1$, and
- instance $\omega' = (n, \mathcal{I}', k, r')$ is feasible, where $\mathcal{I}' = \mathcal{I} \setminus \{[a_i, b_i] : 1 \leq i \leq t - 1\}$ and $r' : \mathcal{I}' \times [k] \mapsto \{0, 1, \dots, n\}$, where $r'(I, c) = r(I, c)$ for $I \in \mathcal{I}'$.

From instance ω' we construct an instance $\hat{\omega} = (n, \hat{\mathcal{I}}, k, \hat{r})$, where set $\hat{\mathcal{I}}$ itself satisfies conditions (a) and (b) from Lemma 4. In particular, intervals in $\hat{\mathcal{I}}$ are disjoint (condition (a)) and therefore feasibility of instance $\hat{\omega}$ can be determined by verifying for every interval $[a, b] \in \hat{\mathcal{I}}$ that $\sum_{j=1}^k \hat{r}([a, b], j) = b - a + 1$.

We define $\hat{\mathcal{I}}$ to be the partition of $\{1, \dots, n\}$ into a minimal number of intervals, such that for each interval $\hat{I} \in \hat{\mathcal{I}}$ and each element $I' \in \mathcal{I}'$ either $\hat{I} \subseteq I'$ or $\hat{I} \cap I' = \emptyset$. We represent $\hat{\mathcal{I}}$ by sequence $([\hat{i}_1, \hat{j}_1], [\hat{i}_2, \hat{j}_2], \dots, [\hat{i}_p, \hat{j}_p])$.

What remains is the assignment of demands to intervals in $\hat{\mathcal{I}}$, i.e. the definition of $\hat{r} : \hat{\mathcal{I}} \times [k] \mapsto \{1, 2, \dots, n\}$. We will show by induction that the definition of $\hat{r}([\hat{a}_1, \hat{b}_1], c)$, for $c \in [k]$, uniquely determines $\hat{r}([\hat{a}_i, \hat{b}_i], c)$, for $2 \leq i \leq p$.

Therefore let us be given values $\hat{r}([\hat{a}_1, \hat{b}_1], c)$, for $c \in [k]$, such that $\sum_{j=1}^k \hat{r}([\hat{a}_1, \hat{b}_1], j) = \hat{b}_1 - \hat{a}_1 + 1$ (otherwise the instance is infeasible and nothing is to show). Assume that values $\hat{r}([\hat{a}_i, \hat{b}_i], c)$, for $c \in [k]$ and $1 \leq i \leq q < p$ have been determined. Consider interval $\hat{I}_{q+1} = [\hat{a}_{q+1}, \hat{b}_{q+1}]$. If $\hat{I}_{q+1} \in \mathcal{I}_1$ or $\hat{I}_{q+1} \in \mathcal{I}_2$ then it clearly must hold $\hat{r}(\hat{\mathcal{I}}, c) = r(\hat{\mathcal{I}}, c)$, for $c \in [k]$. Otherwise, WLOG assume $\hat{a}_{q+1} = a_h$, for some $[a_h, b_h] \in \mathcal{I}_1$, and $\hat{b}_{q+1} = b'_{h'}$, for some $[a'_{h'}, b'_{h'}] \in \mathcal{I}_2$. Let w be such that $[\hat{a}_w, \hat{b}_w] \in \hat{\mathcal{I}}$ and $\hat{a}_w = a'_{h'}$. Then by induction hypothesis all $\hat{r}([a_i, b_i], c)$ with $w \leq i \leq q$ and $c \in [k]$ have been determined and therefore we have to set $\hat{r}([a_{q+1}, b_{q+1}], c) = b'_{h'} - a'_{h'} + 1 - \sum_{i=w}^q \hat{r}([a_i, b_i], c)$.

As for the determination of $\hat{r}([\hat{a}_1, \hat{b}_1], c)$, $c \in [k]$, we have to distinguish between $a_t < a'_1$ and $a_t = a'_1$. Note that $a_t \leq a'_1$ by the definition of index t . In the latter case $\hat{a}_1 = a_t$ and $\hat{b}_1 = \min(b_t, b'_1)$ and thus $\hat{r}([\hat{a}_1, \hat{b}_1], c) = r([a_t, b_t], c)$, respectively $\hat{r}([\hat{a}_1, \hat{b}_1], c) = r([a'_1, b'_1], c)$. In the former case $\hat{a}_1 = a_t$ and $\hat{b}_1 = a'_1$ and we have to try at most $\binom{a'_1 - a_t + k - 1}{k - 1}$ combinations of $\hat{r}([\hat{a}_1, \hat{b}_1], c) \leq r([a_t, b_t], c)$, for $c \in [k]$, such that $\sum_{j=1}^k \hat{r}([\hat{a}_1, \hat{b}_1], j) = a'_1 - a_t$. \square

Corollary 1. *The feasibility problem for given left assignment $P_L = (\mathcal{I}_{P_L}, r_{P_L})$ and right assignment $P_R = (\mathcal{I}_{P_R}, r_{P_R})$ on a set of vertices $V = \{1, 2, \dots, n\}$ can be solved in time $\mathcal{O}(n^k(|\mathcal{I}_{P_L}| + |\mathcal{I}_{P_R}|))$.*

Proof. Let $\mathcal{I}_{P_L} = ([1, u_1], [1, u_2], \dots, [1, u_l])$ and $\mathcal{I}_{P_R} = ([u'_1, n], [u'_2, n], \dots, [u'_r, n])$ be sorted with respect to " \subseteq " and " \supseteq ", respectively, in non-decreasing order, where $l = |\mathcal{I}_{P_L}|$ and $r = |\mathcal{I}_{P_R}|$. Then the feasibility problem on P_L and P_R is equivalent to the feasibility problem on assignments $P'_L = (\mathcal{I}'_{P_L}, r'_{P_L})$ and $P'_R = (\mathcal{I}'_{P_R}, r'_{P_R})$, where $\mathcal{I}'_{P_L} = ([1, u_1], [u_1 + 1, u_2], \dots, [u_{l-1} + 1, u_l])$, $r'_{P_L}([1, u_1], c) = r_{P_L}([1, u_1], c)$, and $r'_{P_L}([u_{i-1} + 1, u_i], c) = r_{P_L}([1, u_i], c) - r_{P_L}([1, u_{i-1}], c)$, for $2 \leq i \leq l$. Symmetrically, $\mathcal{I}'_{P_R} = ([u'_1, u'_2 - 1], [u'_2, u'_3 - 1], \dots, [u'_r, n])$, $r'_{P_R}([u'_r, n], c) = r_{P_L}([u'_r, n], c)$, and $r'_{P_R}([u'_i, u'_{i+1} - 1], c) = r_{P_R}([u'_i, n], c) - r_{P_R}([u'_{i+1}, n], c)$, for $1 \leq i < r$. Interval sets \mathcal{I}'_{P_L} and \mathcal{I}'_{P_R} satisfy conditions (a) and (b) in Lemma 4 and therefore the claim follows. \square

For given left assignment P_L and right assignment P_R on a vertex set $V = \{p, \dots, q\}$, and a ϵ -partial assignment P w.r.t. u^* , procedure REDUCE constructs, considering the recursive call on V_L , a left assignment P'_L and right assignment P'_R on vertex set $V = \{p, \dots, u^*\}$ as follows:

- $\mathcal{I}'_{P_L} = \{[p, t_i] \in \mathcal{I}_{P_L} \mid t_i \leq u^*\}$,
- $\mathcal{I}'_{P_R} = \mathcal{I}'_P \cup \{[s_i, u^*] \mid \exists [s_i, q] \in \mathcal{I}_{P_R} : s_i < u^*\}$, and for all $1 \leq c \leq k$
- $r'_{P_L}(I, c) = r_{P_L}(I, c)$, for $I \in \mathcal{I}'_{P_L}$, and $r'_{P_R}(I, c) = r_P(I, c)$, for $I \in \mathcal{I}'_P$,
- $r'_{P_R}([s_i, u^*], c) = r_{P_R}([s_i, q], c) - r_P([u^*, q], c)$, for $[s_i, q] \in \mathcal{I}_{P_R}$, $s_i < u^*$.

Note that procedure REDUCE on $V_L(u^*)$ or $V_R(u^*)$ results in left assignment P'_L and right assignments P'_R , such that $a_1 = a'_1$, respectively $b_l = b'_m$, holds. Therefore in the construction of instance $\hat{\omega} = (n, \hat{\mathcal{I}}, k, \hat{r})$ in the proof of Lemma 4 function \hat{r} is uniquely determined and we can decide feasibility in time $\mathcal{O}(|V|)$. Adding an ϵ -partial alignment to P'_L and P'_R in step 6 in algorithm MAXCOLORINGAPPROX only splits intervals in $\hat{\mathcal{I}}$ with uniquely determined demands on the new subintervals and thus does not increase the time needed to determine feasibility.

Procedure MAXCOLORINGSPECIAL on a vertex set V is based on the construction of instance $\hat{\omega} = (n, \hat{\mathcal{I}}, k, \hat{r})$ in the proof of Lemma 4. Since intervals in $\hat{\mathcal{I}}$ are disjoint, we can define χ on

Algorithm MaxColoringApprox(\mathcal{I}, V, P_L, P_R):

Input: An instance (\mathcal{I}, V) of MAXCOLORING, and consistent left- and right- assignments P_L and P_R on V

Output: A $(1, 1 + \epsilon)$ approximation (χ, \mathcal{J})

1. **if** $|\mathcal{I}| = 0$, **then**
2. $\chi \leftarrow \text{MAXCOLORINGSPECIAL}(P_L, P_R)$
3. **return** (χ, \emptyset)
4. let $u^* \in V$ be such that $|\mathcal{I}_L(u^*)| \leq m/2$ and $|\mathcal{I}_R(u^*)| \leq m/2$
5. **for** every ϵ -partial assignment P w.r.t. u^* **do**
6. **if** P is consistent with P_L and P_R **then**
7. $(\chi_1, \mathcal{J}_1) \leftarrow \text{MAXCOLORINGAPPROX}(\mathcal{I}_L(u^*), V_L(u^*), \text{REDUCE}(V_L(u^*), P, P_L, P_R))$
8. $(\chi_2, \mathcal{J}_2) \leftarrow \text{MAXCOLORINGAPPROX}(\mathcal{I}_R(u^*), V_R(u^*), \text{REDUCE}(V_R(u^*), P, P_L, P_R))$
9. let $\chi \leftarrow \chi_1 \cup \chi_2$
10. $\mathcal{K} \leftarrow \{I \in \mathcal{I}[u^*] : \frac{r(I, c)}{(1+\epsilon)} \leq r_P(I_{\ell(I, P)}, c) + r_P(I_{j(I, P)}, c) \leq r(I, c)\}$
11. $\mathcal{J} \leftarrow \mathcal{K} \cup \mathcal{J}_1 \cup \mathcal{J}_2$
12. store (χ, \mathcal{J})
13. **return** the recorded solution with largest $w(\mathcal{J})$ value

Figure 2: The dynamic program for computing a $(1, 1 + \epsilon)$ -approximation.

vertices in each interval $\hat{I} \in \hat{\mathcal{I}}$ independently, respecting only $N_\chi(\hat{I}, c) = \hat{r}(\hat{I}, c)$, for all colors $c \in [k]$. Again, due to the specific structure of \mathcal{I}_{P_L} and \mathcal{I}_{P_R} , function \hat{r} is uniquely determined and therefore MAXCOLORINGSPECIAL runs in time $\mathcal{O}(|V|)$.

Let $P = (u^*, \mathcal{I}_P, r_P)$ be an ϵ -partial assignment w.r.t. u^* . Given an interval $I = [s, t] \in \mathcal{I}$, with $u^* \in I$, we let $j(I, P), l(I, P)$ be respectively the smallest and largest indices such that $[u_{j(I, P)}, u'_{l(I, P)}] \subseteq I$, i.e. $j(I, P) = \min\{i : u_i \geq s\}$ and $l(I, P) = \max\{i : u'_i \leq t\}$. If either of these indices does not exist, we set the corresponding $r_P(I_{\ell(I, P)}, c)$ or $r_P(I_{j(I, P)}, c)$ to 0.

Note by (R5) that

$$r_P(I_{\ell(I, P)}, c) + r_P(I_{j(I, P)}, c) \leq N_{\chi'}(I, c) \leq (1 + \epsilon)(r_P(I_{\ell(I, P)}, c) + r_P(I_{j(I, P)}, c)), \quad (14)$$

holds for any coloring $\chi' : V \mapsto [k]$ and ϵ -partial assignment P consistent with χ' .

The algorithm uses two subroutines: MAXCOLORINGSPECIAL checks if a pair of a left- and right-assignments are consistent, and if so, returns a feasible coloring (see Figure 2); REDUCE($V_L(u^*), P, P_L, P_R$) (respectively, REDUCE($V_R(u^*), P, P_L, P_R$)) combines the assignments P, P_L, P_R into a left- and right assignments P'_L, P'_R on $V_L(u^*)$ (respectively, on $V_R(u^*)$).

When the procedure returns, we get two independent colorings $\chi_1 : V_L(u^*) \mapsto [k]$ and $\chi_2 : V_R(u^*) \mapsto [k]$, which are combined into a coloring $\chi = \chi_1 \cup \chi_2$ defined in the obvious way: $\chi(u) = \chi_1(u)$ if $u \in V_L(u^*)$ and $\chi(u) = \chi_2(u)$ if $u \in V_R(u^*)$.

Lemma 5. *Algorithm MAXCOLORINGAPPROX runs in time $T(n, m) = n^{O(\frac{k^2}{\epsilon} \log n \log m)}$.*

Proof. The number of possible ϵ -partial assignments is at most $\mu(n)$, given in (13). This gives the recurrence

$$T(n, m) \leq \text{poly}(n, m) + 2\mu(n) \cdot T\left(\frac{m}{2}\right).$$

The lemma follows. □

Lemma 6. *Algorithm MAXCOLORINGAPPROX returns a coloring $\chi : V \mapsto [k]$ and a subset of intervals $\mathcal{J} \subseteq \mathcal{I}$ such that $w(\mathcal{J}) \geq w(\text{OPT})$ and $r(I, c)/(1 + \epsilon) \leq N_\chi(I, c) \leq (1 + \epsilon)r(I, c)$ for all $I \in \mathcal{J}$ and $c \in [k]$.*

Proof. Let (χ^*, OPT) be an optimal solution. By Lemma 3, there is an ϵ -partial assignment P consistent with χ^* , which will be eventually considered by the algorithm in Step 5. If $I \in \text{OPT}[u^*]$, then $N_{\chi^*}(I, c) = r(I, c)$ and thus (14) implies, for $\chi' = \chi^*$ that I belongs to the set \mathcal{K} selected by the algorithm in Step 10, i.e., $\text{OPT}[u^*] \subseteq \mathcal{K}$, and hence $w(\mathcal{K}) \geq w(\text{OPT}[u^*])$. Since the returned coloring χ is consistent with P , we also know by using $\chi' = \chi$ in (14) that $r(I, c)/(1 + \epsilon) \leq N_\chi(I, c) \leq (1 + \epsilon)r(I, c)$ for $I \in \mathcal{K}$. By induction, we have $w(\mathcal{K}_1) \geq w(\text{OPT}_L(u^*))$, $w(\mathcal{K}_2) \geq w(\text{OPT}_R(u^*))$, $r(I, c)/(1 + \epsilon) \leq N_{\chi_1}(I, c) \leq (1 + \epsilon)r(I, c)$ for $I \in \mathcal{J}_1$, and $r(I, c)/(1 + \epsilon) \leq N_{\chi_2}(I, c) \leq (1 + \epsilon)r(I, c)$ for $I \in \mathcal{J}_2$. The lemma follows. \square

4 Hardness

In this section we show that, in general, deciding whether a feasible coloring exists is NP-hard.

Theorem 3. *The problem of testing the feasibility of an instance of the interval constrained coloring problem is NP-complete when the number of colors is part of the input.*

Proof. Clearly, the problem belongs to NP. To prove the problem is NP-hard we reduce a known NP-hard problem to it using the approach of Chang *et al.* [2]. In the *exact coverage* problem we are given a ground set \mathcal{U} and a collection \mathcal{S} of subsets of \mathcal{U} and we want to know whether there exists a sub-collection $\mathcal{C} \subseteq \mathcal{S}$ of size t , which forms a partition of \mathcal{U} ; that is, $\cup_{S \in \mathcal{C}} S = \mathcal{U}$ and for any $R, S \in \mathcal{C}$ if $R \neq S$ then $R \cap S = \emptyset$. It is well known that exact coverage is NP-complete [5] even when the cardinality of sets in \mathcal{S} is 3.

Let $u = |\mathcal{U}|$ and $s = |\mathcal{S}|$. For the instance of the coloring problem we divide $V = [n]$ into u blocks B_1, \dots, B_u each of length s ; thus, $n = us$ and $B_i = [(i - 1)u + 1, \dots, iu]$. Each color $c \in [k]$ is associated with a specific set S_c in \mathcal{S} ; thus, $k = s$. Let $\mathcal{U} = \{x_1, \dots, x_u\}$ and suppose that x_i is contained r_i in sets. For every $i \in [u]$ we have

$$\begin{aligned} I_i &= [s(i - 1) + 1, \dots, si] & \text{and} & & r(I_i, c) = 1 \text{ for all } c \in [k] \\ I'_i &= [si - t + 1, \dots, s(i + 1) - t] & \text{and} & & r(I'_i, c) = 1 \text{ for all } c \in [k] \\ I''_i &= [in - t - r_i, \dots, in - t + 1] & \text{and} & & r(I''_i, c) = 1 \text{ if and only if } x_i \in S_c \end{aligned}$$

Realize that any coloring satisfying all the I_i and I'_i intervals must use the same set of t colors for the last t positions of every block and the remaining $s - t$ colors for the first $s - t$ position of every block. We therefore encode the cover \mathcal{C} with the last t colors of each block. To enforce that \mathcal{C} is a partition, we ask that for every element $x \in \mathcal{U}$ *exactly* one set in \mathcal{C} contains x . in \mathcal{S} , then we include the interval $I''_i = [in - t - r_i, in - t + 1]$ and require $r(I''_i, c) = 1$ if and only if $x_i \in S_c$. Clearly, a feasible coloring encodes a solution for the exact coverage and vice-versa. It follows that the testing feasibility is NP-hard. \square

Acknowledgments: Thanks to Hubert Chan for useful discussions.

References

- [1] E. Althaus, S. Canzar, M. R. Emmett, A. Karrenbauer, A. G. Marshall, A. Meyer-Basese, and H. Zhang. Computing H/D-exchange speeds of single residues from data of peptic fragments. In *23rd Annual ACM Symposium on Applied Computing*, 2008.

- [2] J. Chang, T. Erlebach, R. Gailis, and S. Khuller. Broadcast scheduling: Algorithms and complexity. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2008.
- [3] K. M. Elbassioni, R. Sitters, and Y. Zhang. A quasi-PTAS for profit-maximizing pricing on line graphs. In *Proceedings of the 15th Annual European Symposium on Algorithms*, pages 451–462, 2007.
- [4] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53(3):324–360, 2006.
- [5] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.
- [6] T. Uno. A fast algorithm for enumerating bipartite perfect matchings. In *Proceedings of the 12th International Conference Algorithms and Computation*, pages 367–379, 2001.