Approximation Algorithms for the Interval Constrained Coloring Problem*

Ernst Althaus† Stefan Canzar† Khaled Elbassioni†

Andreas Karrenbauer† 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 to obtain information about protein tertiary structure. The output of these experiments provides data about the exchange rate of residues in overlapping segments of the protein backbone. These segments 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 segment) 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 and randomized rounding techniques. 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 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 by Althaus *et al.* [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.

---

*An extended abstract appeared in the proceedings of the 11th Scandinavian Workshop on Algorithm Theory, SWAT 2008, held in Gothenburg, Sweden in July 2008.

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

‡Research supported by an Alexander von Humboldt fellowship.

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 with 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 identical 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 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. 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. We refer to this problem as the *interval constrained coloring* problem.

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 $r : \mathcal{I} \times [k] \to Z^+$ be a requirement function such that $\sum_{c \in [k]} r(I, c) = |I|$ for all $I \in \mathcal{I}$. A coloring $\chi : V \to [k]$ is said to be *feasible* if for every $I \in \mathcal{I}$ we have

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

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.

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

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

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

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 was 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 wasleft 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, 3] 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 not 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.* [3] to round x to an integral solution $\hat{x}$.

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 $\hat{x}_{i,c} = 1$ and $\hat{x}_{i,d} = 0$ 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 at least $\gamma_k = \frac{k(k+1-H_{k-1})}{(k+1)!}$.

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 $B_1^c, B_2^c, \ldots, B_{b_c}^c$: For color c and $j = 2, \ldots, b_c - 1$

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

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 $1 < j < b_c$. 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,

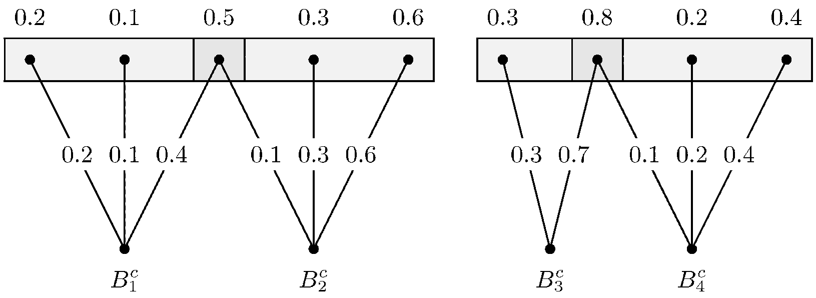

Figure 1: How the blocks B_j^c are constructed. The $x_{i,c}$ values appear on top and the $y_{i,(c,j)}$ values appear on the edges. Note that a block can only overlap with its predecessor or successor. In this case $\alpha_c = 0.7$.

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

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

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

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

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 constraints (6) and (7), each internal block will force a different position in I to be colored c. One 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. ☐

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.* [3] 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 the 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 $\beta_c = \sum_{i\in I\cap B_f^c} y_{i,(c,f)}$, or, equivalently, $\sum_{i\in I\cap B_l^c} y_{i,(c,l)} = 1 - \beta_c$.

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 $\sum_{i\in I\cap B_l^c} \hat{y}_{i,(c,l)} = 1$ respectively. Let $\beta = (\beta_1,\ldots,\beta_k)$ be the vector of offsets for the color classes. For brevity's sake we denote $\Pr[\,\xi \mid \beta\,]$ with $\Pr_\beta[\,\xi\,]$.

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

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

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

As a warm-up we first show that the probability that all requirements for I are fulfilled is at least $\frac{1}{(k+1)!}$. Denote with τ the event $\forall c : \sum_{i\in I} \hat{x}_{i,c} = r(I,c)$. 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.

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

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 $(\frac{1}{2},\ldots,\frac{1}{2}) \in D$. Therefore, setting $D' = [0,\frac{1}{2}]^k$ we get

$$\Pr[\,\tau\,] \geq 2^{k+1} \int_{D'} \max\left\{0, \tfrac{1}{2} - \textstyle\sum_{c\in[k]} \beta_c\right\}\, \mathrm{d}\beta_1\cdots\mathrm{d}\beta_k.$$

The right hand side of the above inequality can be interpreted as the volume of a $(k+1)$-dimensional simplex.

$$\begin{aligned}\Pr[\,\tau\,] &\geq 2^{k+1}\ \mathrm{Vol}\left(\lambda \in R_+^{k+1}\ \middle|\ \textstyle\sum_{i\in[k+1]} \lambda_i \leq \frac{1}{2}\right)\\ &= 2^{k+1}\frac{(\frac{1}{2})^{k+1}}{(k+1)!}\\ &= \frac{1}{(k+1)!}\end{aligned}$$

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 kth 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.

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

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.

$$\Pr[\,\tau\,] \geq k\,2^{k+1}\int_0^{\frac{1}{2}}\left[\int_{[0,z]^{k-1}} \max\left\{0, \tfrac{1}{2}-\textstyle\sum_{c\in[k-1]}\beta_c\right\}\mathrm{d}\beta_1\cdots\mathrm{d}\beta_{k-1}\right]\mathrm{d}z$$

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

$$\Pr[\,\tau\,] \geq k\,2^{k+1}\int_0^{\frac{1}{2}} T(z)\ \mathrm{d}z \tag{11}$$

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 $\mathrm{Vol}\left(\lambda \in R_+^k \;\middle|\; \sum_{i=1}^k \lambda_i \leq \rho\right)$ and recall that $V(\rho) = \frac{\rho^k}{k!}$. Consider what happens when $z \in \left[\frac{1}{4}, \frac{1}{2}\right)$; 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

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

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

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

Plugging (12) into (11) we get

$$\begin{aligned}
\Pr[\,\tau\,] &\geq 2^{k+1}k \left(\int_0^{\frac{1}{2}} V(\tfrac{1}{2})\,\mathrm{d}z + \sum_{i=1}^{k-1} \binom{k-1}{i} (-1)^i \int_0^{\frac{1}{2i}} V(\tfrac{1}{2} - iz)\,\mathrm{d}z \right) \\
&= 2^{k+1}k \left(\int_0^{\frac{1}{2}} \frac{(\frac{1}{2})^k}{k!}\,\mathrm{d}z + \sum_{i=1}^{k-1} \binom{k-1}{i} (-1)^i \int_0^{\frac{1}{2i}} \frac{(\frac{1}{2} - iz)^k}{k!}\,\mathrm{d}z \right) \\
&= 2^{k+1}k \left(\frac{1}{k!2^k} z \Big|_0^{\frac{1}{2}} + \sum_{i=1}^{k-1} \binom{k-1}{i} (-1)^i \; \frac{(\frac{1}{2} - iz)^{(k+1)}}{(k+1)!(-i)} \Big|_0^{\frac{1}{2i}} \right) \\
&= 2^{k+1}k \left(\frac{1}{k!2^{(k+1)}} + \sum_{i=1}^{k-1} \binom{k-1}{i} \frac{(-1)^i}{(k+1)!\,2^{k+1}\,i} \right) \\
&= \frac{k}{(k+1)!} \left(k+1 + \sum_{i=1}^{k-1} \binom{k-1}{i} \frac{(-1)^i}{i} \right)
\end{aligned}$$

Using induction on k, it is straightforward to show that the sum in the last line adds up exactly to $-H_{k-1}$, which gives us the desired bound of γ_k. □

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 $\mathcal{Q}$ [4].

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 $(\chi, \mathcal{I}')$ 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 [5]. 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. (Assignment) *Let $V' = \{p, p+1, \ldots, q\}$. An* assignment *on V' is a pair $\mathscr{A} = (\mathcal{I}, r)$ of intervals $\mathcal{I}$ on V' and a function $r : \mathcal{I} \times [k] \mapsto \{0, 1, \ldots, |V'|\}$ such that*

(C1) $r(I,c) \leq r(I',c)$ for all $I, I' \in \mathcal{I}$ with $I \subseteq I'$ and all $c \in [k]$, and

(C2) $\sum_{c \in [k]} r(I,c) = |I|$ for every $I \in \mathcal{I}$.

$\mathscr{A}$ is called a left-assignment *(respectively,* right-assignment*) if all intervals in $\mathcal{I}$ start at p (respectively, end at q).*

Definition 2. (ϵ-Partial Assignment) *Let $u^* \in V'$ be a given vertex of $V' = \{p, p+1, \ldots, q\}$. A set of $h_1 + h_2 + 2$ intervals $\mathcal{I} = \mathcal{I}_A \cup \mathcal{I}_B$, $\mathcal{I}_A = \{I_1, \ldots, I_{h_1}, I_{h_1+1}\}$ and $\mathcal{I}_B = \{I'_1, \ldots, I'_{h_2}, I'_{h_2+1}\}$, together with a function $r : \mathcal{I} \times [k] \mapsto \{0, 1, \ldots, |V|\}$, such that*

- *(R1) all intervals start or end at u^*: $I_j = [u_j, u^*]$ for $j \in \{1, 2, \ldots, h_1\}$, $I_{h_1+1} = [p, u^*]$, $I'_j = [u^*, u'_j]$ for $j \in \{1, 2, \ldots, h_2\}$, and $I'_{h_2+1} = [u^*, q]$, where $u_{h_1} < u_{h_1-1} < \cdots < u_1 < u^* < u'_1 < u'_2 < \cdots < u'_{h_2}$,*
- *(R2) $(\mathcal{I}_A, r)$ is a right-assignment on $\{p, \ldots, u^*\}$, $(\mathcal{I}_B, r)$ is a left-assignment on $\{u^*, \ldots, q\}$*
- *(R3) for every $I \in \mathcal{I} \setminus \{I_{h_1+1}, I'_{h_2+1}\}$ there exists a color $c \in [k]$ and an integer $i \in \mathbb{Z}_+$ such that $r(I, c) = \lceil (1+\epsilon)^i \rceil$, and*
- *(R4) for every color $c \in [k]$ and integer $i \in \mathbb{Z}_+$ with $i \leq \lfloor (\log r(I_{h_1+1}, c) / \log(1+\epsilon) \rfloor$, there exists $I \in \mathcal{I}_A$ such that $r(I, c) = \lceil (1+\epsilon)^i \rceil$; likewise, for every $c \in [k]$ and $i \in \mathbb{Z}_+$ with $i \leq \lfloor (\log r(I'_{h_2+1}, c) / \log(1+\epsilon) \rfloor$, there exists $I' \in \mathcal{I}_B$ such that $r(I', c) = \lceil (1+\epsilon)^i \rceil$.*

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

From property (*C2*) of an assignment (with $|I| \leq n$) and property (*R3*) of an ϵ-partial assignment it follows $h_1, h_2 \leq \lceil k \log n / \log(1+\epsilon) - 1 \rceil$. In Figure 2 vertices $\{p, p+1, \ldots, u^*\} \subseteq V'$ are shown along with four intervals from $\mathcal{I}_A$, all ending at u^* (*R1*). Note that intervals I_{j_1}, I_{j_2} and I_{j_h} satisfy condition (*R4*) for color $c \in [k]$, since $r(I_{j_1}, c) = \lceil (1+\epsilon)^1 \rceil$, $r(I_{j_2}, c) = \lceil (1+\epsilon)^2 \rceil$ and $r(I_{j_h}, c) = \lceil (1+\epsilon)^h \rceil$, for $h = \lceil (\log r(I_{h_1+1}, c) / \log(1+\epsilon) - 1 \rceil$.

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 $n^{h_1+h_2}$ possible choices for the vertices $u_1, u_2, \ldots, u_{h_1}, u'_1, u'_2, \ldots, u'_{h_2}$. For each interval $I \in \mathcal{I}$, the number of non-negative integer requirements $r(I, c)$, $c \in [k]$, satisfying (*R3*) is $\binom{|I|+k-1}{k-1} \leq \left(1 + |I|(\ln k + 1)/(k-1)\right)^{k-1}$. Letting $i_1 = |I_1|$, $i_j = |I_j \setminus I_{j-1}|$, for $2 \leq j \leq h_1 + 1$, and similarly $i'_1 = |I'_1| - 1$, $i'_j = |I'_j \setminus I'_{j-1}|$, for $2 \leq j \leq h_2 + 1$, we observe by

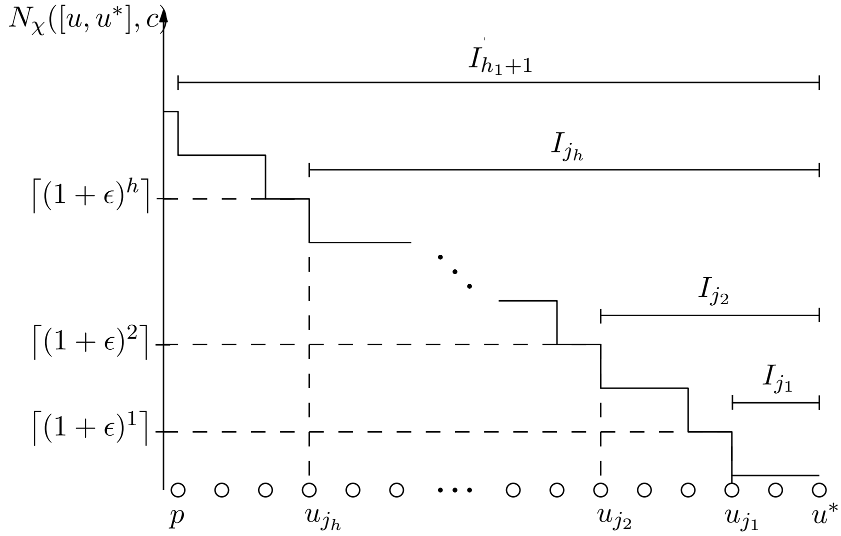

Figure 2: The number of vertices in interval $[u, u^*]$ colored $c \in [k]$ by χ is monotonically increasing on $u - p$. According to (*R5*), an ϵ-partial assignment consistent with χ has to contain intervals I_{j_1}, I_{j_2} and I_{j_h} with $r(I_{j_1}, c) = \lceil(1+\epsilon)^1\rceil$, $r(I_{j_2}, c) = \lceil(1+\epsilon)^2\rceil$ and $r(I_{j_h}, c) = \lceil(1+\epsilon)^h\rceil$, for $h = \lceil(\log r(I_{h_1+1}, c)/\log(1+\epsilon) - 1\rceil$. Interval $I_{h_1+1} = [p, u^*]$, see (*R1*).

(*R2*) and (*R3*) that

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

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

Definition 3. (Consistent Assignment) *Let $\chi : V \mapsto [k]$ be a coloring of V. We say that an assignment $\mathscr{A} = (\mathcal{I}, r)$ is* consistent *with χ if $N_\chi(I, c) = r(I, c)$ for all $c \in [k]$ and*

$I \in \mathcal{I}$. *Two assignments* $\mathscr{A}_1$ *and* $\mathscr{A}_2$ *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* $\mathscr{P} = (u^*, \mathcal{I}, r)$ *on* V' *w.r.t.* u^* *that is consistent with* χ.

Proof. Assume that $V' = \{p, p+1, \ldots, q\}$. 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 $\mathscr{P}$ as follows. Let $u'_0 = u^*$. For $j = 1, 2, \ldots, h_2$ let

$$u'_j = \min\Big\{u > u'_{j-1} \mid \exists\, i \in \mathbb{Z}_+, c \in [k] :\ N_\chi([u^*, u], c) = \lceil (1+\epsilon)^i \rceil \Big\}. \qquad (14)$$

The highest index j for which such an $u'_j < q$ exists determines the value of h_2. In accordance with condition (*R1*), we set $I'_j = [u^*, u'_j]$ for $j = 1, 2, \ldots, h_2$ and $I_{h_2+1} = [u^*, q]$. In a similar way we define h_1 and the intervals I_j for $j = 1, 2, \ldots, h_1 + 1$ (see Figure 2). Finally, we define $r(I, c) = N_\chi(I, c)$ for all $c \in [k]$ and $I \in \mathcal{I}$, which naturally satisfies (*R2*) and (*R3*). The definition of interval endpoints according to (14) guarantees (*R4*) and (*R5*). ☐

Observation 1. *Let* $\mathscr{P} = (u^*, \mathcal{I}_{\mathscr{P}}, r_{\mathscr{P}})$ *be an* ϵ*-partial assignment w.r.t.* u^*. *Given an interval* $I = [s, t] \in \mathcal{I}$ *with* $u^* \in I$, *we let* $j(I, \mathscr{P})$ *and* $\ell(I, \mathscr{P})$ *be, respectively, the smallest and largest indices such that* $[u_{j(I,\mathscr{P})}, u'_{\ell(I,\mathscr{P})}] \subseteq I$, *i.e.* $j(I, \mathscr{P}) = \min\{i :\ u_i \geq s\}$ *and* $\ell(I, \mathscr{P}) = \max\{i :\ u'_i \leq t\}$ *(see Figure 3). If either of these indices does not exist, we set the corresponding value of* $r_{\mathscr{P}}(I_{\ell(I,\mathscr{P})}, c)$ *or* $r_{\mathscr{P}}(I_{j(I,\mathscr{P})}, c)$ *to* 0. *Then by property (R5) of an* ϵ*-partial assignment*

$$r_{\mathscr{P}}(I_{\ell(I,\mathscr{P})}, c) + r_{\mathscr{P}}(I_{j(I,\mathscr{P})}, c) \leq N_{\chi'}(I, c) \leq (1+\epsilon)\big(r_{\mathscr{P}}(I_{\ell(I,\mathscr{P})}, c) + r_{\mathscr{P}}(I_{j(I,\mathscr{P})}, c)\big) \quad (15)$$

holds for any $c \in [k]$ *and coloring* $\chi' : V \mapsto [k]$ *such that* $\mathscr{P}$ *is consistent with* χ.

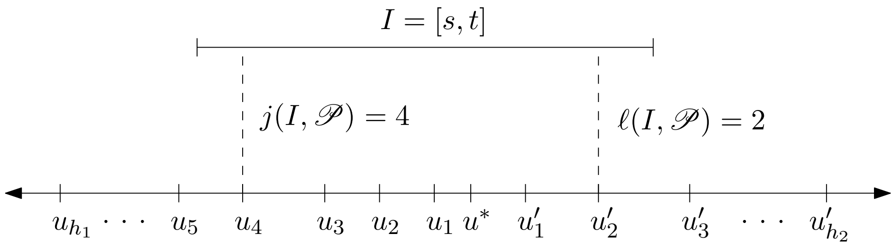

Figure 3: For an ϵ-partial assignment $\mathscr{P}$ w.r.t. u^* and a given interval $I \in \mathcal{I}$, $j(I, \mathscr{P})$ and $\ell(I, \mathscr{P})$ are defined to be the smallest and largest indices, respectively, such that $[u_{j(I,\mathscr{P})}, u'_{\ell(I,\mathscr{P})}] \subseteq I$.

3.2 A Divide-and-Conquer Algorithm

The pseudocode describing our dynamic-programming algorithm is presented below as a procedure called MaxColoringApprox, which takes as parameters an instance $(n, k, \mathcal{I}, r)$ of problem MaxColoring and consistent left- and right-assignments $\mathscr{A}_l$ and $\mathscr{A}_r$. To compute an $(1, 1+\epsilon)$-approximation, we set $\mathscr{A}_l$ and $\mathscr{A}_r$ to be empty in the initial call.

The algorithm is based on a divide-and-conquer paradigm where a vertex u^* in the middle of V is picked and all intervals containing u^* are evaluated to determine whether they should be taken into the solution. To do this evaluation conservatively, the procedure iterates over all ϵ-partial assignments $\mathscr{P}$ w.r.t. to the middle vertex u^* that are consistent with $\mathscr{A}_l$ and $\mathscr{A}_r$, then recurses on the subsets of intervals to the left and right of u^*.

Procedure MaxColoringApprox uses two subroutines: MaxColoringSpecial checks whether a pair of a left- and a right-assignment is consistent, and if so, returns a feasible coloring; Reduce $(V_L(u^*), \mathscr{P}, \mathscr{A}_l, \mathscr{A}_r)$ $\big($Reduce $(V_R(u^*), \mathscr{P}, \mathscr{A}_l, \mathscr{A}_r)\big)$ combines the assignments $\mathscr{P}, \mathscr{A}_l$ and $\mathscr{A}_r$ into left- and right-assignments $\mathscr{A}_l', \mathscr{A}_r'$ on $V_L(u^*)$ $\big($respectively, on $V_R(u^*)\big)$. For a more detailed description of the two subroutines see below.

Algorithm 1: MaxColoringApprox(V, $\mathcal{I}$, $\mathscr{A}_l$, $\mathscr{A}_r$)

Data: An instance $(n, k, \mathcal{I}, r)$ of MaxColoring
Result: An $(1, 1+\epsilon)$-approximation $(\chi, \mathcal{J})$
1 **if** $|\mathcal{I}| = 0$ **then**
2     $\chi \leftarrow$ MaxColoringSpecial($\mathscr{A}_l, \mathscr{A}_r$)
3   **return** $(\chi, \emptyset)$
4 let $u^* \in V$ be such that $|\mathcal{I}_L(u^*)| \le m/2$ and $|\mathcal{I}_R(u^*)| \le m/2$
5 **forall** *ϵ-partial assignments* $\mathscr{P} = (u^*, \mathcal{I}_{\mathscr{P}}, r_{\mathscr{P}})$ **do**
6   **if** *$\mathscr{P}$ is consistent with $\mathscr{A}_l$ and $\mathscr{A}_r$* **then**
7         $(\chi_1, \mathcal{J}_1) \leftarrow$ MaxColoringApprox$\big(V_L(u^*), \mathcal{I}_L(u^*),$ Reduce$(V_L(u^*), \mathscr{P}, \mathscr{A}_l, \mathscr{A}_r)\big)$
8         $(\chi_2, \mathcal{J}_2) \leftarrow$ MaxColoringApprox$\big(V_R(u^*), \mathcal{I}_R(u^*),$ Reduce$(V_R(u^*), \mathscr{P}, \mathscr{A}_l, \mathscr{A}_r)\big)$
9         $\chi \leftarrow \chi_1 \cup \chi_2$
10         $\mathcal{K} \leftarrow \{I \in \mathcal{I}(u^*) : \frac{r(I,c)}{(1+\epsilon)} \le r_{\mathscr{P}}(I_{\ell(I,\mathscr{P})}, c) + r_{\mathscr{P}}(I_{j(I,\mathscr{P})}, c) \le r(I,c) \quad \forall c\}$
11         $\mathcal{J} \leftarrow \mathcal{K} \cup \mathcal{J}_1 \cup \mathcal{J}_2$
12   store $(\chi, \mathcal{J})$
13 **return** the recorded solution with largest $w(\mathcal{J})$ value

From the recursive calls in lines **7** and **8** we obtain two independent colorings $\chi_1 : V_L(u^*) \mapsto [k]$ and $\chi_2 : V_R(u^*) \mapsto [k]$, which are combined in line **9** 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 $u \in V_R(u^*)$.

Given a left-assignment $\mathscr{A}_l = (\mathcal{I}_l, r_l)$ and right-assignment $\mathscr{A}_r = (\mathcal{I}_r, r_r)$ on a vertex set $V' = \{p, \ldots, q\}$ and an ϵ-partial assignment $\mathscr{P} = (u^*, \mathcal{I}_A \cup \mathcal{I}_B, r_{\mathscr{P}})$, procedure Reduce constructs, considering the recursive call on $V'_L(u^*)$ in line **7**, a left-assignment $\mathscr{A}'_l = (\mathcal{I}'_l, r'_l)$ and right-assignment $\mathscr{A}'_r = (\mathcal{I}'_r, r'_r)$ on vertex set $V'' = \{p, \ldots, u^*\}$ by cutting intervals at u^* as follows (see Figure 4):

- $\mathcal{I}'_l = \{[p,t] \in \mathcal{I}_l \mid t \leq u^*\}$,
- $\mathcal{I}'_r = \mathcal{I}_A \cup \{[s,u^*] \mid \exists\, [s,q] \in \mathcal{I}_r : s < u^*\}$,
- $r'_l(I,c) = r_l(I,c)$ for $I \in \mathcal{I}'_l$ and $r'_r(I,c) = r_{\mathscr{P}}(I,c)$ for $I \in \mathcal{I}_A$, for all $c \in [k]$,
- $r'_r([s,u^*],c) = r_r([s,q],c) - r_{\mathscr{P}}([u^*,q],c) + 1$ for $[s,q] \in \mathcal{I}_r$, $s < u^*$, for all $c \in [k]$.

In the recursive call in line **8** procedure Reduce combines the given assignments according to a symmetric schema. Notice that $\mathcal{I}_l = \emptyset$ in the leftmost and $\mathcal{I}_r = \emptyset$ in the rightmost path of the recursion tree.

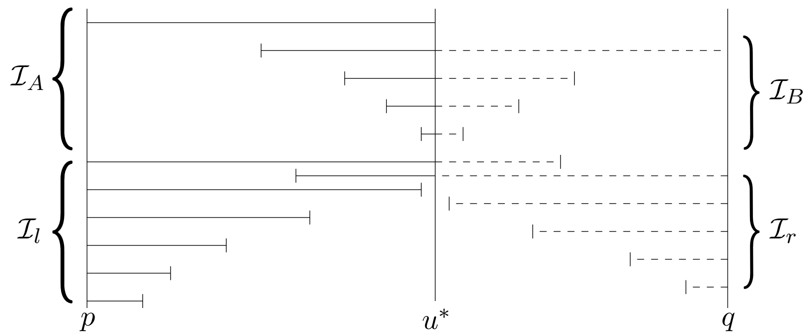

Figure 4: For given left-assignment $\mathscr{A}_l = (\mathcal{I}_l, r_l)$, right-assignment $\mathscr{A}_r = (\mathcal{I}_r, r_r)$ and an ϵ-partial assignment $\mathscr{P} = (u^*, \mathcal{I}_A \cup \mathcal{I}_B, r_{\mathscr{P}})$, in the recursive call on $V_L(u^*)$ procedure Reduce cuts intervals on the vertical line at index u^* such that the new left- and right-assignments $\mathscr{A}'_l$ and $\mathscr{A}'_r$ contain the intervals shown by solid lines. Interval $[p,q]$, contained both in $\mathcal{I}_l$ and $\mathcal{I}_r$, is omitted.

In the following Lemma 4 and Theorem 3 we show how procedure MaxColoringSpecial can check consistency of assignments $\mathscr{A}_l = (\mathcal{I}_l, r_l)$ and $\mathscr{A}_r = (\mathcal{I}_r, r_r)$ on vertex set V in line **2** in time $\mathcal{O}(|\mathcal{I}_l| + |\mathcal{I}_r|)$. Note that sets $\mathcal{I}_l$ and $\mathcal{I}_r$ each contain an interval spanning all vertices in V. This is due to intervals I_{h_1+1} and I'_{h_2+1} in Definition 2 of an ϵ-partial assignment and due to the specific structure of the assignments constructed by procedure Reduce.

Lemma 4. *Let $\mathscr{A} = (\mathcal{I}, r)$ be an assignment on $V = \{1, 2, \ldots, n\}$ where set $\mathcal{I}$ can be partitioned into two sets $\mathcal{I}_1$ and $\mathcal{I}_2$, such that for $p \in \{1,2\}$ it holds*

(P1) $I_i \cap I_j = \emptyset$, $\forall I_i, I_j \in \mathcal{I}_p$, i.e. intervals are disjoint and

(P2) $\bigcup_{I \in \mathcal{I}_p} I = [1,n]$, i.e. the intervals span all vertices.

Then it can be decided in time $\mathcal{O}(|\mathcal{I}|)$ whether a feasible coloring $\chi : V \mapsto [k]$ for $\mathscr{A}$ exists, i.e. a coloring χ such that $\mathscr{A}$ is consistent with χ.

Proof. We represent interval set $\mathcal{I}_1$ as sequence $\big([s_1,t_1],[s_2,t_2],\dots,[s_l,t_l]\big)$ and set $\mathcal{I}_2$ as sequence $\big([\bar{s}_1,\bar{t}_1],[\bar{s}_2,\bar{t}_2],\dots,[\bar{s}_m,\bar{t}_m]\big)$, where $s_i = t_{i-1}+1$ for $2 \le i \le l$, and similarly $\bar{s}_i = \bar{t}_{i-1}+1$ for $2 \le i \le m$. Property $(P2)$ implies $s_1 = \bar{s}_1 = 1$ and $t_l = \bar{t}_m = n$. For $1 \le i \le l$ we denote $[s_i,t_i]$ by I_i and for $1 \le i \le m$ we denote $[\bar{s}_i,\bar{t}_i]$ by $\bar{I}_i$.

From assignment $\mathscr{A}$ we construct an equivalent assignment $\mathscr{A}' = (\mathcal{I}', r')$, where intervals in $\mathcal{I}'$ are disjoint and therefore feasibility of $\mathscr{A}'$ can be determined by verifying for every interval $[s,t] \in \mathcal{I}'$ that

$$\sum_{c\in[k]} r'([s,t],c) = t - s + 1.$$

We define $\mathcal{I}'$ to be the partition of $\{1,2,\dots,n\}$ into a minimal number of intervals, such that for each interval $I' \in \mathcal{I}'$ and each element $I \in \mathcal{I}$ either $I' \subseteq I$ or $I' \cap I = \emptyset$ (see Figure 5(a)). We represent $\mathcal{I}'$ by sequence $\big([s'_1,t'_1],[s'_2,t'_2],\dots,[s'_r,t'_r]\big)$ and again denote $[s'_i,t'_i]$ by I'_i for $1 \le i \le r$.

What remains is the assignment of requirements to intervals in $\mathcal{I}'$, i.e. the definition of $r' : \mathcal{I}' \times [k] \mapsto \{1,2,\dots,n\}$. We will define function r' recursively, i.e. for $c \in [k]$ the value $r'(I'_i,c)$ might depend on values $r'(I'_j,c)$ for $j < i$. Due to the minimality of $\mathcal{I}'$, $t'_1 = \min(t_1,\bar{t}_1)$ and interval I'_1 will coincide with either I_1 or $\bar{I}_1$. In Figure 5(a) the latter case holds. Therefore any coloring χ feasible for assignment $\mathscr{A}$ will satisfy (1) for interval I'_1 if and only if $r'(I'_1,c) = r(I_1,c)$ or $r'(I'_1,c) = r(\bar{I}_1,c)$, respectively, for all $c \in [k]$. Now consider an interval I'_i for arbitrary $2 \le i \le r$. If $I'_i \in \mathcal{I}_1$ or $I'_i \in \mathcal{I}_2$, as e.g. $I'_4 \in \mathcal{I}_2$ in Figure 5(a), for assignment $\mathscr{A}'$ to be equivalent with assignment $\mathscr{A}$ it must hold $r'(I'_i,c) = r(I'_i,c)$, for all $c \in [k]$. Otherwise, without loss of generality assume $s'_i = s_q$ for some $I_q \in \mathcal{I}_1$ and $t'_i = \bar{t}_{q'}$ for some $\bar{I}_{q'} \in \mathcal{I}_2$. Let p be such that $I'_p \in \mathcal{I}'$ and $s'_p = \bar{s}_{q'}$ (see Figure 5(b)). If we assume that any coloring χ feasible for $\mathscr{A}$ satisfies (1) for all intervals I'_j with $1 \le j \le i-1$, then χ will satisfy (1) for interval I'_i if and only if

$$r'(I'_i,c) = r(\bar{I}_{q'},c) - \sum_{j=p}^{i-1} r'(I'_j,c), \text{ for all } c \in [k]. \tag{16}$$

□

Clearly the above lemma can be generalized to the case where $\mathcal{I}$ can be partitioned into an arbitrary number of sets, each satisfying conditions $(P1)$ and $(P2)$.

Theorem 3. *Let $V = \{1,2,\dots,n\}$. For given left assignment $\mathscr{A}_l = (\mathcal{I}_l, r_l)$ with $[1,n] \in \mathcal{I}_l$ and right assignment $\mathscr{A}_r = (\mathcal{I}_r, r_r)$ with $[1,n] \in \mathcal{I}_r$, it can be decided in time $\mathcal{O}(|\mathcal{I}_l| + |\mathcal{I}_r|)$ whether $\mathscr{A}_l$ and $\mathscr{A}_r$ are consistent.*

Proof. Let $\mathcal{I}_l = \big([1,t_1],[1,t_2],\dots,[1,t_p]\big)$ with $t_p = n$ and $\mathcal{I}_r = \big([s_1,n],[s_2,n],\dots,[s_q,n]\big)$ with $s_1 = 1$ be sorted with respect to "$\subseteq$" and "$\supseteq$", respectively, in non-decreasing order. Then assignments $\mathscr{A}_l$ and $\mathscr{A}_r$ are consistent if and only if the following assignments $\mathscr{A}'_l = (\mathcal{I}'_l, r'_l)$ (see Figure 6) and $\mathscr{A}'_r = (\mathcal{I}'_r, r'_r)$ are consistent:

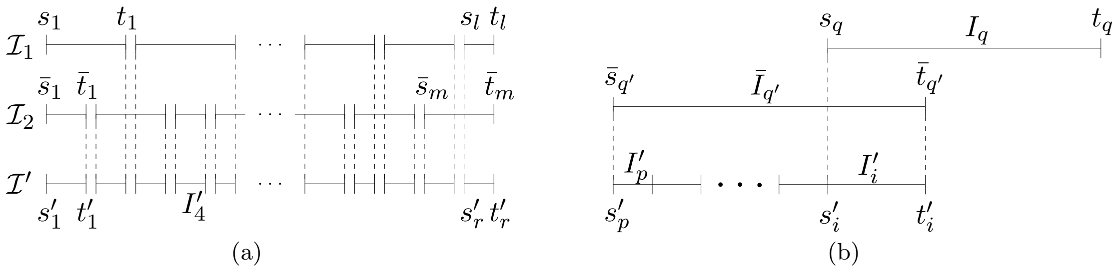

Figure 5: (a) Set $\mathcal{I}_1$ and $\mathcal{I}_2$ satisfy (*P1*) and (*P2*) in Lemma 4. For each interval $I' \in \mathcal{I}'$ and each element $I \in \mathcal{I}$, $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$, either $I' \subseteq I$ or $I' \cap I = \emptyset$. (b) In the construction of an equivalent assignment $\mathscr{A}'$ in the proof of Lemma 4 the number of vertices that have to be colored c in interval I'_i is obtained by equation (16).

- $\mathcal{I}'_l = \big([1, t_1], [t_1 + 1, t_2], \ldots, [t_{p-1} + 1, t_p]\big)$,
- $r'_l([1, t_1], c) = r_l([1, t_1], c)$ and $r'_l([t_{i-1} + 1, t_i], c) = r_l([1, t_i], c) - r_l([1, t_{i-1}], c)$, for $2 \leq i \leq p$ and $c \in [k]$.
- $\mathcal{I}'_r = \big([s_1, s_2 - 1], [s_2, s_3 - 1], \ldots, [s_q, n]\big)$ and
- $r'_r([s_q, n], c) = r_r([s_q, n], c)$ and $r'_r([s_i, s_{i+1} - 1], c) = r_r([s_i, n], c) - r_r([s_{i+1}, n], c)$, for $1 \leq i < q$ and $c \in [k]$.

Interval sets $\mathcal{I}'_l$ and $\mathcal{I}'_r$ satisfy conditions (*P1*) and (*P2*) in Lemma 4 and therefore the claim follows. ◻

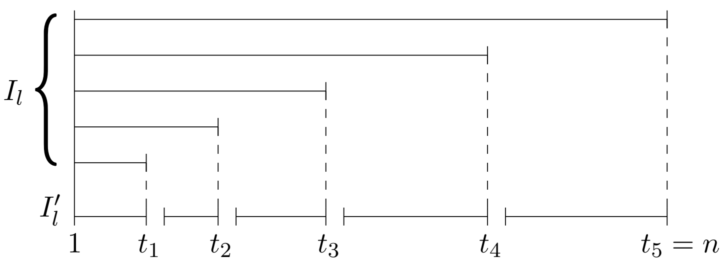

Figure 6: Left-assignment $\mathscr{A}_l = (\mathcal{I}_l, r_l)$ can be transformed into an equivalent assignment $\mathscr{A}'_l = (\mathcal{I}'_l, r'_l)$. For every interval in $I'_l \backslash \{[1, t_1]\}$ its requirement r'_l is equal to the difference between the requirements r_l of its defining intervals in I_l (see proof of Theorem 3).

Since intervals in $\mathcal{I}'$ of assignment $\mathscr{A}'$ in the proof of Lemma 4 are disjoint, procedure MaxColoringSpecial can determine coloring χ in line **2** on vertices in each interval $I' \in \mathcal{I}'$ independently, respecting only $N_\chi(I', c) = r'(I', c)$ for all colors $c \in [k]$. Therefore procedure MaxColoringSpecial runs in time $\mathcal{O}(|V|)$.

In line **6** of procedure MaxColoringApprox consistency of an ϵ-partial assignment $\mathscr{P} = (u^*, \mathcal{I}_A \cup \mathcal{I}_B, r_{\mathscr{P}})$ and left- and right-assignments $\mathscr{A}_l$ and $\mathscr{A}_r$ has to be verified. From the definition of an ϵ-partial assignment (see Definition 2) it follows that $(\mathcal{I}_A, r_{\mathscr{P}})$ forms a left-assignment on $V_L(u^*)$ and $(\mathcal{I}_B, r_{\mathscr{P}})$ a right-assignment on $V_R(u^*)$, where every vertex is spanned by at least one interval. As such, similar as in the proof of Theorem 3, they can be transformed into equivalent assignments containing only disjoint intervals. As intervals in $\mathcal{I}_A$ and $\mathcal{I}_B$ only intersect in u^*, this transformation results in a single set of intervals $\widetilde{\mathcal{I}}$ satisfying conditions (*P1*) and (*P2*) in Lemma 4. As shown above in the description of procedure MaxColoringSpecial, checking consistency of assignments $\mathscr{A}_l$ and $\mathscr{A}_r$ can be reduced to a feasibility problem of an assignment $\mathscr{A}' = (\mathcal{I}', r')$ (see proof of Lemma 4), where $\mathcal{I}'$ itself satisfies (*P1*) and (*P2*) in Lemma 4. In summary, consistency of $\mathscr{P}$, $\mathscr{A}_l$ and $\mathscr{A}_r$ can be verified in line **6** by applying Lemma 4 to sets $\widetilde{\mathcal{I}}$ and $\mathcal{I}'$ in time $\mathcal{O}(|V|)$.

Theorem 4. *For $|V| = n$ and $|\mathcal{I}| = m$, 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 $\mu(n)$, bounded in (13). This gives the recurrence

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

The theorem follows. □

Theorem 5. *Algorithm* MaxColoringApprox *returns a coloring $\chi : V \mapsto [k]$ and a subset of intervals $\mathcal{J} \subseteq \mathcal{I}$ such that $w(\mathcal{J}) \ge w(\mathrm{OPT})$ and $r(I,c)/(1+\epsilon) \le N_\chi(I,c) \le (1+\epsilon)r(I,c)$ for all $I \in \mathcal{J}$ and $c \in [k]$.*

Proof. Let (χ^*, OPT) be an optimal solution to an instance of the MaxColoring problem. By Lemma 3, there is an ϵ-partial assignment $\mathscr{P}$ consistent with χ^*, which will be eventually considered by the algorithm in line **5**. If $I \in \mathrm{OPT}[u^*]$, then $N_{\chi^*}(I,c) = r(I,c)$ for all $c \in [k]$ and thus (15) implies, for $\chi' = \chi^*$, that I belongs to the set $\mathcal{K}$ selected by the algorithm in line **10**, i.e. $\mathrm{OPT}[u^*] \subseteq \mathcal{K}$, and hence $w(\mathcal{K}) \ge w(\mathrm{OPT}[u^*])$. Since $\mathscr{P}$ is consistent with the coloring χ obtained in line **9**, we also know, by using $\chi' = \chi$ in (15), that

$$r(I,c)/(1+\epsilon) \le N_\chi(I,c) \le (1+\epsilon)r(I,c) \text{ for } I \in \mathcal{K}.$$

By induction, we have $w(\mathcal{J}_1) \ge w\big(\mathrm{OPT}_L(u^*)\big)$ and $w(\mathcal{J}_2) \ge w\big(\mathrm{OPT}_R(u^*)\big)$. Furthermore, we know that $r(I,c)/(1+\epsilon) \le N_{\chi_1}(I,c) \le (1+\epsilon)r(I,c)$ for $I \in \mathcal{J}_1$ and $r(I,c)/(1+\epsilon) \le N_{\chi_2}(I,c) \le (1+\epsilon)r(I,c)$ for $I \in \mathcal{J}_2$. The theorem follows. □

4 Hardness

In this section we show that, in general, deciding whether a feasible coloring exists is NP-hard. Furthermore, one can show that problem MaxColoring is APX-hard when $k = 2$ (see [6]) using a similar technique as in [7], where the authors show APX-hardness for the *maximum feasible subsystem* problem with the constraint matrix being a clique.

Theorem 6. *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 [8] 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 $B_1, \ldots, B_u$ each of length s; thus, $n = us$ and $B_i = [(i-1)s + 1, \ldots, i\,s]$. Each color $c \in [k]$ is associated with a specific set S_c in $\mathcal{S}$; thus, $k = s$. Let $\mathcal{U} = \{x_1, \ldots, x_u\}$ and suppose that x_i is contained in r_i sets. For every $i \in [u]$ we have

$$I_i = [s\,(i-1) + 1,\ s\,i] \qquad \text{and} \qquad r_{I_i,c} = 1 \text{ for all } c \in [k]$$
$$I''_i = [s\,i - t - r_i + 2,\ s\,i - t + 1] \qquad \text{and} \qquad r_{I''_i,c} = 1 \text{ if and only if } x_i \in S_c$$

and for every $i \in [u-1]$ we have

$$I'_i = [s\,i - t + 1,\ s\,(i+1) - t] \qquad \text{and} \qquad r_{I'_i,c} = 1 \text{ for all } c \in [k]$$

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 partition $\mathcal{C}$ with the last t colors of each block. To enforce $\mathcal{C}$ to be a partition, i.e. every element $x \in \mathcal{U}$ *exactly* one set in $\mathcal{C}$ contains x in $\mathcal{S}$, we include the interval $I''_i = [s\,i - t - r_i + 2, s\,i - t + 1]$ 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 testing feasibility is NP-hard. ☐

Acknowledgments: Thanks to Hubert Chan for useful discussions.

References

[1] Ernst Althaus, Stefan Canzar, Mark R. Emmett, Andreas Karrenbauer, Alan G. Marshall, Anke Meyer-Basese, and Huimin Zhang. Computing H/D-exchange speeds of single residues from data of peptic fragments. In *23rd Annual ACM Symposium on Applied Computing*, 2008.

[2] Jessica Chang, Thomas Erlebach, Renars Gailis, and Samir Khuller. Broadcast scheduling: Algorithms and complexity. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2008.

[3] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53(3):324–360, 2006.

[4] Takeaki Uno. A fast algorithm for enumerating bipartite perfect matchings. In *Proceedings of the 12th International Conference Algorithms and Computation*, pages 367–379, 2001.

[5] Khaled M. Elbassioni, René Sitters, and Yan 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.

[6] S. Canzar. *Lagrangian Relaxation - Solving NP-hard problems in Computational Biology via Combinatorial Optimization.* PhD thesis, Universität des Saarlandes, 2008.

[7] Khaled Elbassioni, Rajiv Raman, Saurabh Ray, and René Sitters. On the approximability of the maximum feasible subsystem problem with 0/1 coefficients. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2009. (accepted for publication).

[8] Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.