A Polynomial Delay Algorithm for Enumerating Approximate Solutions to the Interval Constrained Coloring Problem

STEFAN CANZAR, Center for Computational Biology, McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine. Baltimore, Maryland, USA.
KHALED ELBASSIONI, Masdar Institute of Science and Technology. Abu Dhabi, UAE.
JULIÁN MESTRE, School of Information Technologies, University of Sydney. Australia.

We study the INTERVAL CONSTRAINED COLORING problem, a combinatorial problem arising in the interpretation of data on protein structure emanating from experiments based on hydrogen/deuterium exchange and mass spectrometry. The problem captures the challenging task of increasing the spatial resolution of experimental data in order to get a better picture of the protein structure. Since solutions proposed by any algorithmic framework have to ultimately be verified by biochemists, it is important to provide not just a single solution, but a valuable set of candidate solutions. Our contribution is a polynomial-delay polynomial-space algorithm for enumerating all exact solutions plus further approximate solutions, which are guaranteed to be within an absolute error of two of the optimum within fragments of the protein, i.e. within sets of consecutive residues. Our experiments indicate that the quality of the approximate solutions is comparable to the optimal ones, in terms of deviation from the underlying true solution. In addition, the experiments also confirm the effectiveness of the method in reducing the delay between two consecutive solutions considerably, compared to what it takes an integer programming solver to produce the next exact solution.

Categories and Subject Descriptors: G.1.6 [**Numerical Analysis**]: Optimization—*Linear programming*; *Integer programming*; G.2.1 [**Discrete Mathematics**]: Combinatorics—*Combinatorial algorithms*; *Permutations and combinations*; G.4 [**Mathematical Software**]: Algorithm design and analysis; J.3 [**Life and Medical Sciences**]: Biology and genetics

General Terms: Algorithms

Additional Key Words and Phrases: LP rounding, hydrogen-deuterium exchange, protein structure

ACM Reference Format:
Stefan Canzar, Khaled Elbassioni, and Julián Mestre. 2013. A Polynomial Delay Algorithm for Enumerating Approximate Solutions to the Interval Constrained Coloring Problem. *ACM J. Exp. Algor.* V, N, Article A (January YYYY), 20 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

A challenging and central problem in computational biology is to determine the tertiary structure of proteins, their dynamics, and their interactions. Experimental techniques such as X-ray crystallography and nuclear magnetic resonance [Zuiderweg 2002] provide the highest resolution of the protein structure. However, both methods require large (milligram) quantities of the protein under consideration. Other techniques [Leite and

This work is supported in part by NIH, under grant R01 HG006677.
A preliminary version of this paper appeared in Proceedings of the Twelfth Workshop on Algorithm Engineering and Experiments, ALENEX 2010 [Canzar et al. 2010].

DOI:http://dx.doi.org/10.1145/0000000.0000000

Cascio 2002; Sharp et al. 2004] also suffer from physical limitations or can possibly change the conformation of the protein.

Solution-phase hydrogen/deuterium exchange (sHDX) is an alternative approach that can work at low concentration and high molecular weight [Englander 2006], and does not change the conformation of the protein. In a typical experiment, the protein is diluted in heavy water (D_2O) causing labile hydrogens in the protein to exchange place with deuteriums from the D_2O. This exchange takes place at a higher rate at sites exposed to the solvent. To obtain local instead of global exchange information the protein or protein complex is digested by enzymes called proteases, that is, it is decomposed to polypeptides. Since deuterium is heavier than hydrogen, its incorporation over time (rate) can be monitored by the increase in mass of each peptic fragment using mass spectrometry (MS). For a schematic diagram of the sHDX experiment, see Figure 1. From the obtained exchange rates one can draw conclusions concerning the solvent accessibility of amide protons within the original protein construct and thus derive information about its structure.

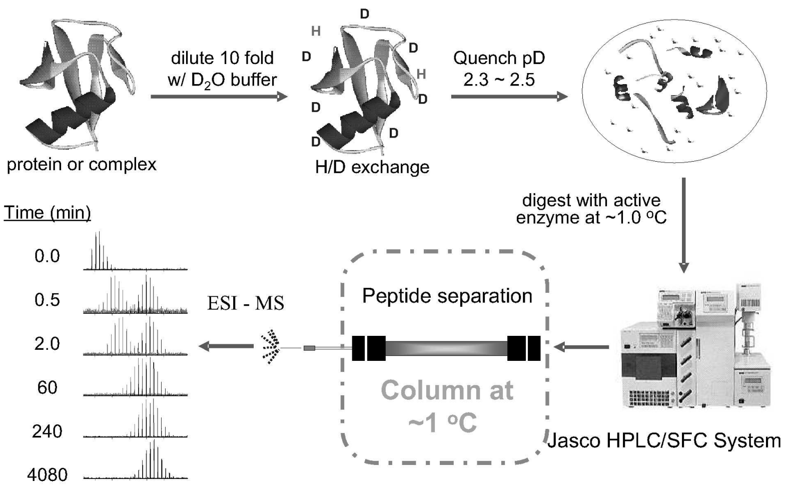


Fig. 1. Schematic diagram of HDX experiment. The exchange, initiated by diluting the protein solution 10 fold into deuterated buffer, is quenched by lowering the temperature and dropping the pH for predetermined time points. The protein is then digested and injected onto the HPLC (High-performance liquid chromatography) for separation and mass analysis.

One downside of this approach is that the data obtained after post-processing using the maximum entropy method (MEM) [Zhang et al. 1997] is at the level of the polypeptides generated from the protease digest of the protein (see Figure 2). In other words, instead of knowing the exchange rate of individual residues, we only get aggregate information of the exchange rate of fragments of the protein. The INTERVAL CONSTRAINED COLORING problem captures the combinatorial problem of inferring the individual rates given this coarse resolution data. The problem has received much attention in the last two years [Althaus et al. 2010; Althaus et al. 2009; Althaus et al. 2011; Althaus et al. 2008; Byrka et al. 2010; Canzar 2008; Komusiewicz et al. 2011].

No	Amino acid sequence	All	Slow	Medium	Fast
1	$\underline{GLSDGEWQQVLNVWGKVEADIAGHGQEVL}$	28	15	8	5
2	$\underline{GLSDGEWQQVL}$	10	7	2	1
3	$\underline{NVWGKVEAD}$	8	5	2	1
4	$\underline{VLNVWGKVEADIAGHGQE}$	17	12	1	4
5	$\underline{NVWGKVEA}$	7	5	1	1
6	$\underline{WQQVLNVWGKVEADIAGHGQEVL}$	15	11	1	3
7	$\underline{GLSDGEW}$	6	4	1	1
8	$\underline{WQQVL}$	4	3	1	0
9	$\underline{IAGHGQEVL}$	8	7	1	0

Fig. 2. Overlapping peptic fragments as seen in an sHDX of the model protein myoglobin. Each line corresponds to a polypeptide resulting from the protease digest of the protein. On the right it shows the total number of amide hydrogens exchanged in the peptide (All) and the number of amide hydrogens predicted to be either slow, medium or fast by maximum entropy method (MEM) evaluation of the H/D exchange rate distribution [Zhang et al. 1997].

1.1. Mathematical abstraction and contribution

The INTERVAL CONSTRAINED COLORING problem is 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 to as colors from now on. 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. The colors represent the different exchange rates in the experimental data. Each interval is a fragment of the protein chain. The requirements are defined by the aggregate information of the exchange rates of residues contained in each fragment.

Our main contribution is an efficient procedure for enumerating all possible feasible solutions, plus possibly some near feasible solutions. Note that in the context of interpreting data obtained by sHDX/MS experiments, our solutions have to be verified by biochemists in a second step. While solutions that are (near-)optimal in a given mathematical model are indistinguishable from a mathematical point of view, they still might be of different biological relevance. Our algorithm is based on a backtracking approach commonly used in enumeration problems (for instance, to enumerate all vertices of a given $0/1$-polytope [Bussieck and Lübbecke 1998]) and the rounding approach suggested in [Althaus et al. 2011]. Finally, we show through experimental evaluation that our method compares favorably to previously known algorithms.

1.2. Related work and comparison to our work

Althaus et al. [2010] formulated the problem of improving the resolution of exchange data as a linear minimization problem subject to integer linear constraints (ILP) and described an algorithm that often assigns deuterium exchange rates with single amino acid resolution. Based on this formulation, a branch and bound-based approach was given to enumerate all possible exchange rates for single residues. The approach proposed in [Althaus et al. 2009] integrates the MEM analysis and the improvement of the spatial resolution of the data into a single ILP formulation. More recently, Althaus et al. [2011] studied this problem from a theoretical point of view. They showed it to be NP-hard in general and developed an approximation algorithm that, given a feasible instance, finds a coloring that satisfies each coloring requirement within ± 1 of the prescribed value. In fact, the enumeration algorithm we present in this paper uses an extension of this approximation method. They also developed a quasi-polynomial-time approximation algorithm for a variant of the problem which requires to find a coloring satisfying as many fragments as possible. However, they only provide algorithms for producing a single solution, and do not provide any experimental evaluation. Very recently, Komusiewicz *et al.* [Komusiewicz et al. 2011] showed that the problem is fixed-parameter tractable with respect to parameters such as the maximum fragment length, and the maximum number of fragments containing a given residue. The problem of finding a solution minimizing the error incurred has been shown [Byrka et al. 2010] to be NP-hard even when three exchange rates are used. In contrast, when only two rates are used, the problem is polynomially solvable [Althaus et al. 2010].

While the approach of Althaus et al. [2010] is somewhat related to ours, there are some fundamental differences. First, our algorithm does not only produce exact solutions, but may also produce approximate solutions, in which case each component of the approximate solution is guaranteed to be within an absolute error of 2 of the optimal error[1] for each fragment and color. The complexity of the problem of finding colorings that satisfy all the requirements within ± 1 of the prescribed value remains open in the integral case. Since the exact solutions are produced by minimizing the total error in the LP formulation, and the requirements are inherently noisy, an additional error of 2 is acceptable, specially in the cases when the requirements for each fragment and color are large. Furthermore, this method has the advantage that we still get some solutions even in the case when there is a gap between the cost of the best fractional and the best integral solution; as we shall see from the experiments, this can happen frequently for large instances. More generally, given any target error parameter $h \in \mathbb{N}$ our algorithm can enumerate efficiently all solutions in which each component is within h of the optimal LP error, plus possibly some solutions which are within $h + 2$ of the optimal LP error (we emphasize that the algorithm outputs *different* solutions).

Second, we do not rely on any *integer* programming solver, only *linear* programming is used. In fact, we will show that our enumeration algorithm has the *polynomial delay* property, which means that the time elapsed between two consecutive outputs is polynomial in the *input size*. More specifically, our bound will depend only *linearly* on the number of colors, in contrary to the intuition that such bound should be exponential. Note that such a polynomial delay procedure is very useful when it is required to output a certain number k of solutions, since this can be done in time polynomial in the input size and linear in k. We remark that no such bounds were given on the algorithm of Althaus et al. [2010]. Furthermore, the rounding procedure itself is very efficient, since it relies on computing a perfect matching in convex bipartite graphs [Glover 1967]. In effect, as our experiments confirm, for larger size instances, it might take very long (maybe a few days), to produce a single exact solution (as the solution is obtained by solving an NP-hard integer programming problem), where as an approximate (but still reasonably good) solution is always guaranteed to be output by our method (typically in much less than a second).

The rest of the paper is organized as follows. In the next section, we formally define the problem and state the natural ILP formulation. In Section 3, we extend the rounding technique of Althaus et al. [2011], which was suggested for a different ILP formulation where we use a binary variable for each residue-color pair, to the ILP formulation used here where we partition residues into segments and use an integral variable for each segment-color pair. As we shall see below, this extension is useful for significantly reducing the number of solutions output by the algorithm. In Section 4, we give the

[1] We point out the inaccuracy in the absolute error (claimed to be 1) in an earlier version of this paper [Canzar et al. 2010].

$$\text{minimize} \sum_{I\in\mathcal{I},\ i\in[k]} e(I,i) \qquad (2)$$

$$\begin{array}{lrll}
\text{subject to} & & & \\
& \sum_{i\in[k]} x_{s,i} = |s| & \forall\, s \in \mathcal{S} & (3)\\
& r(I,i) - \sum_{s\in\mathcal{S}(I)} x_{s,i} \le e(I,i) & \forall\, I \in \mathcal{I}, i \in [k] & (4)\\
& -r(I,i) + \sum_{s\in\mathcal{S}(I)} x_{s,i} \le e(I,i) & \forall\, I \in \mathcal{I}, i \in [k] & (5)\\
& x_{s,i} \in \mathbb{Z}_+ & \forall\, s \in \mathcal{S}, i \in [k] & (6)
\end{array}$$

Fig. 3. ILP formulation for the INTERVAL CONSTRAINED COLORING problem.

enumeration algorithm. Finally, a thorough experimental evaluation of our algorithm is presented in Section 5.

2. FORMAL PROBLEM DEFINITION

Let $\mathcal{I}$ be a set of intervals defined on the set $V = [n]$, let $[k]$ be a set of color classes, and let $r : \mathcal{I} \times [k] \to \mathbb{N}$ be a requirement function such that $\sum_{i\in[k]} r(I,i) = |I|$ for all $I \in \mathcal{I}$. Rather than working with the elements of V itself, as noted in [Althaus et al. 2010], in order to reduce the number of output solutions it is better to define the coloring problem over segments of residues rather than singletons. A *segment* $s \subseteq [n]$ is defined as a maximal sub-interval of V in the interior of which no interval from $\mathcal{I}$ begins or ends. By this definition, the segments are disjoint and naturally ordered from left to right. Let $\mathcal{S}$ denote the set of all segments in the given instance, and let $\mathcal{S}(I)$ be the set of segments contained in iteral $I \in \mathcal{I}$. A *coloring* $\chi : \mathcal{S} \times [k] \to \mathbb{N}$ is an assignment such that for every $s \in \mathcal{S}$ we have $\sum_{i=1}^{k} \chi(s,i) = |s|$. Given this information, we would like to find a coloring χ that minimizes the error

$$\sum_{I\in\mathcal{I},\ i\in[k]} |r(I,i) - \sum_{s\in\mathcal{S}(I)} \chi(s,i)|. \qquad (1)$$

The problem is captured by the integer program given in Figure 3. We note that this formulation was first introduced by Althaus et al. [2010]. The integer variable $x_{s,i}$ indicates the number of residues in segment s, which are assigned color $i \in [k]$. Constraint (3) enforces that each segment gets as many colors as residues in it and constraints (4) and (5) enforce that every requirement is satisfied within the error bound $e(I,i)$. The objective is to minimize the total error incurred by the coloring.

Let $\mathcal{P}$ be the polytope obtained by relaxing the integrality constraint (6) in the above integral problem. That is, $\mathcal{P}$ is the set of vectors of (x,e) obeying (3), (4), (5) and $x_{s,i} \ge 0$ for all $s \in \mathcal{S}$ and $i \in [k]$.

For a positive integer h, and an error vector e^*, we call a coloring $\chi : \mathcal{S} \times [k] \to \mathbb{N}$ *h-approximate of type* 0, with respect to e^*, if

$$|r(I,i) - \sum_{s\in\mathcal{S}(I)} \chi(s,i)| \le e^*(I,i) + h, \quad \text{for all } I \in \mathcal{I} \text{ and } i \in [k].$$

On the other hand, χ will be called *h-approximate of type* 1, with respect to e^*, if there exists $e' : \mathcal{I} \times [k] \rightarrow \mathbb{R}_+$ such that

$$\sum_{I \in \mathcal{I},\ i \in [k]} e'(I,i) \leq \sum_{I \in \mathcal{I},\ i \in [k]} e^*(I,i),$$

and

$$|r(I,i) - \sum_{s \in \mathcal{S}(I)} \chi(s,i)| \leq e'(I,i) + h, \text{ for all } I \in \mathcal{I} \text{ and } i \in [k].$$

Usually, e^* will be taken to be an optimal solution of the LP relaxation of the above ILP. As we will see in the next section, there is an efficient algorithm that, given a fractional solution, returns an integral solution where no requirement is violated by more than two units; that is, the rounding routine returns a 2-approximate solution of type 0. Solutions of type 1 represent a generalization of this concept, where the overall error must be preserved, but we have the freedom to redistribute it in any way among the requirements.

3. A PLUS TWO GUARANTEE

Let (x, e) be a fractional solution in $\mathcal{P}$. We generalize the scheme of Althaus et al. [2011] to round x to an integral solution $\hat{x}$ with the following properties:

THEOREM 3.1. *Given a fractional solution $(x, e) \in \mathcal{P}$ we can construct in polynomial time an integral solution $\hat{x}$ with the following properties*

(P1) For every $s \in \mathcal{S}$ we have $\sum_{i \in [k]} \hat{x}_{s,i} = |s|$.
(P2) For every $I \in \mathcal{I}$ and $i \in [k]$ we have $|\sum_{s \in \mathcal{S}(I)} \hat{x}_{s,i} - r(I,i)| \leq \lceil e(I,i) \rceil + 1$.
(P3) For every $s \in \mathcal{S}$ and $i \in [k]$, if $x_{s,i} \in \mathbb{Z}_+$ then we have $x_{s,i} = \hat{x}_{s,i}$.

The first property (P1) implies that each segment gets one color for each residue it contains. The second property (P2) implies that each error incurred on this coloring requirement is increased by at most two. In other words, there always exists a 2-approximate coloring of type 0, with respect to the given error vector e, that can be found in polynomial time. Property (P3) will be needed by the enumeration algorithm, and is easy to achieve by first fixing $x_{s,i} = \hat{x}_{s,i}$ for every $s \in \mathcal{S}$ and $i \in [k]$ such that $x_{s,i} \in \mathbb{Z}_+$ (which is equivalent to coloring $x_{s,i}$ residues in segment s with color i), and modifying the requirements accordingly.

We now describe an algorithm that, given a fractional solution $(x, e) \in \mathcal{P}$, produces an integral solution $\hat{x}$ with properties (P1) and (P2). To this end, we define blocks $B^i_1, B^i_2, \ldots, B^i_{b_i}$. For color $i \in [k]$ let $b_i = \lceil \sum_s x_{s,i} \rceil$; then for each $j = 1, \ldots, b_i$ define

$$B^i_j = \left\{ s \in \mathcal{S} \ : \ j-1 < \sum_{\substack{r \in \mathcal{S} \\ r \leq s}} x_{r,i} \ \wedge \ j > \sum_{\substack{r \in \mathcal{S} \\ r < s}} x_{r,i} \right\} \tag{7}$$

where the segment orders $\leq$ and $<$ follow the natural left-to-right ordering of the protein. Furthermore, we will say that the block *starts* at the smallest segment under this ordering, and *ends* at the largest segment under this ordering.

Each block is a contiguous set of segments containing at least[2] one fractional unit of color i. We will assign the value of each variable $x_{s,i}$ to those blocks B^i_j such that

[2] Althaus et al. [2011] use blocks of size one and assume there is no error, that is, $e(I,i) = 0$ for all $I \in \mathcal{I}$ and $i \in [k]$.

$$\mathcal{Q} = \left\{ \quad y \quad : \quad \begin{array}{rr} \sum_{i\in[k]} \sum_{j: s \in B^i_j} y_{s,(i,j)} = |s| & \forall\, s \in \mathcal{S} \\ \sum_{s\in B^i_j} y_{s,(i,j)} = 1 & \forall\, i \in [k] \text{ and } j < b_i \\ \sum_{s \in B^i_{b_i}} y_{s,(i,b_i)} \leq 1 & \forall\, i \in [k] \\ y_{s,(i,j)} \geq 0 & \forall\, s \in \mathcal{S}, i \in [k], j \in [b_i] \end{array} \right\}$$

Fig. 4. The polytope of the capacitated assignment problem.

$s \in B^i_j$. For each segment s that belongs to B^i_j we define a variable $y_{s,(i,j)}$, which will represent how much of $x_{s,i}$ is assigned to B^i_j. We ask that $x_{s,i} = \sum_{j:s\in B^i_j} y_{s,(i,j)}$ and $\sum_{s\in B^i_j} y_{s,(i,j)} = 1$ for every $j < b_i$. It is not difficult to see that if s belongs to a single block B^i_j of color i then we must set $y_{s,(i,j)} = x_{s,i}$. Otherwise, the segment s belongs to two adjacent blocks B^i_j and B^i_{j+1} where B^i_j ends at s and B^i_{j+1} starts at s. If B^i_j also starts at s then $y_{s,(i,j)} = 1$, otherwise $y_{s,(i,j)} = j - \sum_{r<s} x_{r,i}$. Similarly, if B^i_{j+1} ends at s then $y_{s,(i,j+1)} = 1$, otherwise $y_{s,(i,j+1)} = \sum_{r\leq s} x_{r,i} - (j-1)$.

Notice that y defines a fractional assignment between the set of segments $\mathcal{S}$ and the set of blocks. More specifically, every block (except the last of each color) is assigned fractionally to the segments that make it up and each segment s is assigned $|s|$ many blocks. Let $\mathcal{Q}$ be the polytope of this capacitated assignment problem. The formal definition of the polytope is given in Figure 4. It is well known that the matrix defining the constraints of such an assignment problem is totally unimodular (see e.g. [Schrijver 2003, Chapter 18]), and thus $\mathcal{Q}$ is integral. Therefore, any fractional solution $y \in \mathcal{Q}$ can be turned into an integral solution $\hat{y} \in \mathcal{Q}$ without violating any constraints; furthermore, this can even be done in polynomial time. This implies that the assignment problem defined by $\mathcal{Q}$ is feasible.

We can now state the procedure to round (x, e) to an integral coloring. First, using x, create the blocks as described above. Then set up the capacitated assignment problem defined by $\mathcal{Q}$ and solve it using a standard polynomial-time algorithm to get a solution $\hat{y}$. We note that for any block, the set of neighbouring segments form a contiguous set; therefore a simple greedy algorithm [Glover 1967] solves our assignment problem in *linear time*. Finally we construct the solution $\hat{x}$ from $\hat{y}$ by setting

$$\hat{x}_{s,i} = \sum_{j:s\in B^i_j} \hat{y}_{s,(i,j)}. \tag{8}$$

It remains to show that $\hat{x}$ has properties (P1) and (P2). We prove each property separately with the following lemmas.

LEMMA 3.2. *Let $\hat{y}$ be an integral solution for $\mathcal{Q}$ and let $\hat{x}$ be the coloring induced by $\hat{y}$. Then $\sum_{i\in[k]} \hat{x}_{s,i} = |s|$ for every segment $s \in \mathcal{S}$.*

PROOF. From the first constraint of $\mathcal{Q}$ we have

$$\sum_{i\in[k]} \sum_{j:s\in B^i_j} \hat{y}_{s,(i,j)} = |s|$$

and from (8) we infer that

$$\sum_{i\in[k]} \hat{x}_{s,i} = |s|.$$

□

LEMMA 3.3. *Let $\hat{y}$ be an integral solution for $\mathcal{Q}$ and let $\hat{x}$ be the coloring induced by $\hat{y}$. Then $|\sum_{s\in\mathcal{S}(I)} \hat{x}_{s,i} - r(I,i)| \leq \lceil e(I,i)\rceil + 1$ for all $I \in \mathcal{I}$ and $i \in [k]$.*

PROOF. For simplicity, we assume $\sum_{s\in\mathcal{S}(I)} \hat{x}_{s,i} = r(I,i) + e(I,i)$. The case where the number of residues colored i is less than the prescribed requirement can be handled in a similar fashion.

Let $B^i_{b+1}, B^i_{b+2}, \dots, B^i_{b+a}$ be those blocks associated with color i that intersect I. Recall that for each of these blocks, except perhaps the last, we have $\sum_{s\subseteq B^i_{b+j}} y_{s,(i,b+j)} = 1$. Notice that the first and last block may overlap I partially; however, all the remaining $a-2$ blocks must lie strictly inside I. Each of these inner blocks will contribute one residue colored i inside of I. The first and last blocks may contribute up to one additional residue colored i each. It follows that $a \geq r(I,i) + e(I,i)$, or equivalently, since a is an integer, $a \geq r(I,i) + \lceil e(I,i)\rceil$. We argue that $a \leq r(I,i) + \lceil e(I,i)\rceil + 1$. If $e(I,i) = \lceil e(I,i)\rceil$ then it is easy to see that $a \leq r(I,i) + \lceil e(I,i)\rceil + 1$. If $e(I,i) < \lceil e(I,i)\rceil$ by the fact that the $a-2$ internal blocks each contribute one residue colored i we have that $a < r(I,i) + \lceil e(I,i)\rceil + 2$.

Suppose $a = r(I,i) + \lceil e(I,i)\rceil$ and $\lceil e(I,i)\rceil = e(I,i)$, then all blocks lie inside I. Hence, the rounded solution $\hat{y}$ must color with i exactly a residues in I and no additional error is incurred. Now consider the case $a = r(I,i) + \lceil e(I,i)\rceil$ and $\lceil e(I,i)\rceil > e(I,i)$. Therefore, we can guarantee that the number of residues within I colored i is between $r(I,i) + \lceil e(I,i)\rceil - 2 = r(I,i) + \lfloor e(I,i)\rfloor - 1$ and $r(I,i) + \lceil e(I,i)\rceil$. In both cases the lemma statement holds.

Now consider what happens when $a = r(I,i) + \lceil e(I,i)\rceil + 1$. The rounded solution $\hat{y}$ must color with i at least $a-2 = r(I,i) + \lceil e(I,i)\rceil - 1$ and at most $a = r(I,i) + \lceil e(I,i)\rceil + 1$ residues of I. Again, the lemma statement holds. □

4. THE ENUMERATION ALGORITHM

We use a standard backtracking procedure to enumerate all approximate colorings. On a high level, this procedure works by picking a segment s at the current node of the recursion tree and branching on all possible colorings of s. For each such coloring c, a restricted instance is obtained by fixing the colors inside s to c. Then an LP is formulated on the restricted instance with the additional constraint that the error from the resulting instance is "comparable to" the initial error e^* (obtained at the root node). If this LP is feasible, we proceed inductively from that branch. Otherwise, if the LPs at all branches turn out to be infeasible, then we resort to rounding the fractional feasible solution x^* at the current node. A naive implementation of this would give an algorithm with delay polynomial in n and $|\mathcal{I}|$, but *exponential* in k. This is due to the fact that the LP might be infeasible for a large interval of values assumed by one or more variables $x_{s,i}$, and yet we only detect this when we assign the last variable in the segment $x_{s,k}$. In order to reduce the delay to a polynomial in k, we modify this branching procedure, such that we assign colors to variables, *one at a time*. We can further speed it up by using binary search to discover early when the LP is infeasible; see Remark 4.1 below.

A similar procedure was used in [Althaus et al. 2010]. The crucial difference is that when the LP is feasible, while the ILP is not, we can still find an approximate integral solution by the rounding procedure described in Theorem 3.1. More generally, given any parameter $h \in \mathbb{N}$, our algorithm can enumerate efficiently all h-approximate solutions of type 0, together with some (possibly exponentially many) $h+2$-approximate solutions, or alternatively all exact solutions, together with some 2-approximate solutions of type 1.

Algorithm 1 FINDALL$(\chi, \ell, i, \bar{x})$:

Input: (*Global*) A problem instance $(\mathcal{I}, r)$ implying a set of segments $\mathcal{S}$, an initial optimal error vector $\bar{e} : \mathcal{I} \times [k] \to \mathbb{R}_+$, approximation guarantee $h \in \mathbb{N}$, and type of approximation $t \in \{0, 1\}$; (*Local*) a partial coloring $\chi : \mathcal{S} \times [k] \to \mathbb{N}$, the index ℓ of the current segment s_ℓ, the current color index i, and the current feasible fractional vector $\bar{x}$.

Output: A superset of all h-approximate colorings of type t, with respect to $\bar{e}$, that consists of $h + 2$-approximate solutions that are consistent with χ.

```
 1: $q \leftarrow |s_\ell| - \sum_{j=1}^{i-1} \chi(s_\ell, j)$
 2: if $i = k$ then
 3:     $p \leftarrow q$
 4: else
 5:     $p \leftarrow 0$
 6: end if
 7: $foundFeasible \leftarrow$ FALSE
 8: for $j = p$ to $q$ do
 9:     $\chi(s_\ell, i) := j$
10:     if LP$(\mathcal{I}, r, \bar{e}, h, t, \chi)$ has a FEASIBLE solution $x$ then
11:         $foundFeasible \leftarrow$ TRUE
12:         if $i = k$ then
13:             if $\ell = |\mathcal{S}|$ then
14:                 output $\chi$
15:             else
16:                 FINDALL$(\chi, \ell + 1, 1, x)$
17:             end if
18:         else
19:             FINDALL$(\chi, \ell, i + 1, x)$
20:         end if
21:     end if
22: end for
23: if $foundFeasible =$ FALSE then
24:     output APPROXINTEGRAL$(\bar{x})$
25: end if
```

The algorithm is given as Algorithm 1. We assume a numbering of the segments from 1 to $|\mathcal{S}|$ and assign all possible values to the variables in lexicographic order of their indices (see lines 16 and 19) that are not violating constraint (3) (lines 1-6).

The algorithm is initially called with an empty coloring χ, segment index $\ell = 1$, color $i = 1$ and the optimal fractional solution $\bar{x} = x^*$ for the LP relaxation. Besides the problem instance $(\mathcal{I}, r)$ itself which implies a certain subdivision of the residues into segments, it also requires an optimal fractional error $\bar{e} = e^*$ and parameters t and h specifying type and quality of the approximation sought.

LP$(\mathcal{I}, r, \bar{e}, h, t, \chi)$ denotes an LP defined by the constraints (3), (4), (5), the non-negativity constraints, the constraints fixing the variables according to the partial coloring χ, and additionally, the error-bound constraints

$$e(I, i) \leq \bar{e}(I, i) + h \; \forall I \in \mathcal{I} \text{ and } i \in [k], \tag{9}$$

if we seek an $h + 2$-approximate coloring of type $t = 0$, and

$$\sum_{I \in \mathcal{S},\ i \in [k]} e(I, i) \leq \sum_{I \in \mathcal{S},\ i \in [k]} \bar{e}(I, i), \tag{10}$$

if we seek a 2-approximate coloring of type $t = 1$.

The procedure APPROXINTEGRAL(x) takes as an input a fractional point x in the polytope $\mathcal{P}$ and outputs an integral solution having the guarantees stated in Theorem 3.1. In particular the solution output by APPROXINTEGRAL(x) is an extension of χ due to Property (P3).

At the current node of the recursion tree, the LP is checked for feasibility for all possible extensions of the current partial coloring on s_ℓ (line 9) that do not violate constraint (3). If the LP is infeasible then there is nothing to do. Otherwise there are two possible ways to proceed. Either all variables have been fixed already and thus we have found an integral solution (line 14) or the next free segment/color is selected and the procedure recurses (lines 16, 19). If no feasible extension of χ on s_ℓ has been found and thus no integral solution extending the current coloring exists, we call the rounding procedure of Theorem 3.1 (line 24) on the current feasible fractional vector $\bar{x}$.

Remark 4.1. The search procedure can be speeded up by performing binary search on the values of the variables $x_{i,s}$. Specifically, this can be done by keeping integral upper and lower bounds $b'_{s,i}$ and $b''_{s,i}$ for each variable $x_{s,i}$. Let s and i be the current segment and color index, respectively. To proceed, the procedure reduces the current interval $[b'_{s,i} : b''_{s,i}]$ on the variable $x_{s,i}$ into two intervals $[b'_{s,i} : m]$ and $[m+1 : b''_{s,i}]$, where $m = \lfloor (b'_{s,i} + b''_{s,i})/2 \rfloor$, and recurses on the two corresponding problems. If the two problems are infeasible, then there are no integral solutions extending the current coloring. In this case, we call the rounding procedure of Theorem 3.1. Otherwise, we continue on the feasible path(s). Eventually, we reach a node at which $b'_{s,i} = b''_{s,i}$, and hence the current partial coloring χ can be extended by setting $\chi_i(s) = b'_{s,i}$. For the sake of simplicity of presentation, we will not incorporate such binary search into the algorithm.

The quality of the performance of the algorithm is given by the following theorem.

THEOREM 4.2. *Algorithm* FINDALL *has the following properties:*

(i) When called with $t = 0$, the algorithm outputs a superset of all type 0 h-approximate colorings. Any non h-approximate coloring output by the algorithm is an $h+2$-approximate coloring of type 0.

(ii) When called with $t = 1$, the algorithm outputs a superset of all optimal colorings. Any non-optimal coloring output by the algorithm is a 2-approximate coloring of type 1.

(iii) The algorithm requires only $O(n|\mathcal{I}|+k|\mathcal{S}|)$ space and the delay between two successive output is $O(k|\mathcal{S}|\log(s_{max}+1)T + |\mathcal{I}| + n)$, where s_{max} is the maximum length of a segment and T is the maximum time needed to solve an LP instance in step 10.

PROOF.

(i) It can be easily seen by induction (on the depth of the node) that, at any node of the recursion tree $\mathbf{T}$ with input $(\chi, \ell, i, \bar{x})$, the following two invariants hold: (C1) $\bar{x}$ is feasible for LP$(\mathcal{I}, r, \bar{e}, h, t, \chi)$, and (C2) $\bar{x}$ is consistent with χ, i.e., $\bar{x}$ and χ agree on the components fixed by χ. (Indeed, both invariants are initially true since χ is initially empty and $(\bar{x}, \bar{e}) = (x^*, e^*)$ is an optimal solutions of the initial LP relaxation, and they continue to hold by the definition of LP$(\mathcal{I}, r, \bar{e}, h, t, \chi)$ and the fact that we check for feasibility of this LP in step 10 before creating a new node in $\mathbf{T}$.)

Note that the outputs are only produced at leaf nodes. At each such node, either every segment in $\mathcal{S}$ is completely colored, or any possible *integral* extension of the current partial coloring χ results in an infeasible LP. In the former case, we output the coloring $\mathcal{X}$, which by (C1) is an integral feasible solution, and hence is an h-approximate coloring of type 0. In the latter case, we invoke the procedure for approximate integral solution, which may incur an additional error of ± 2 for each interval and color, according to Theorem 3.1, thus yielding an $h+2$-approximate coloring of type 0.

It remains to argue that any h-approximate coloring of type 0 must be output by the algorithm. Suppose that χ' is such a coloring. Consider a node $\mathbf{v}$ in the recursion tree $\mathbf{T}$ with input $(\chi, \ell, i, \bar{x})$, such that χ is consistent with χ' (the root of the recursion tree trivially satisfies this condition). Then there is a selection of j in line 8 such that $j = \chi'(s_\ell, i)$, and hence the current coloring χ will be extended by setting $\chi(s_\ell, i) = \chi'(s_\ell, i)$ in line 9. This implies that the LP in line 10 is feasible. Thus there a child node $\mathbf{u}$ of $\mathbf{v}$ in $\mathbf{T}$, such that the input partial coloring χ'' at $\mathbf{u}$ is also consistent with χ'. This inductive argument shows that the algorithm will output χ' at line 14 at some leaf of $\mathbf{T}$.

(ii) Can be proved using an argument similar to (i).

(iii) We notice first that the outputs produced by the algorithm are *different*: recall that we only round solutions at leaf nodes, and that the rounding procedure has the property that the integral components of the fractional solution are not changed (Property (P3)). Thus any pair of solutions must differ in at least one variable assignment $x_{s,i}$. Furthermore, since the algorithm essentially performs a depth first search on the recursion tree, the maximum time needed until a new output is produced is bounded by a constant times the depth of the tree, times the maximum time needed at each node of the tree. As can be easily seen, the depth of the tree is no more than the number of segments $|\mathcal{S}|$ times the number of colors k. At each node, exactly one LP is solved for each value of j in the loop in line 8, and any feasible LP implies that a new solution is produced. The number of iterations of the loop is at most s_{max}. This would give a bound linear in s_{max}. However by using binary search this can be reduced to $\log s_{max}$; see remark 4.1 above. Furthermore, at most one rounding step can be done before any new output is produced. The cost for rounding is $O(n + |\mathcal{I}|)$ as one can easily verify. This implies the stated bound on the delay. Since the algorithm does not need to store the outputs, it can be implemented with polynomial linear in the size of the recursion tree (plus the space to store $\mathcal{I}$).

□

5. EXPERIMENTAL EVALUATION

We have implemented our polynomial-delay enumeration algorithm in C++ using CPLEX 12.2[3] (default settings) with Concert Technology. The first set of experiments compares our implementation PolyDelayHDX with the two existing approaches for enumerating all exact solutions, a branching approach by Althaus et al. [2010] and a method based on Lagrangian relaxation [Canzar 2008]. For the purpose of comparability we set $h = 0$ and consider type 1 approximations. In this case, provided the LP relaxation is integral (as is the case for the real-word instances, see below), the set of solutions output by PolyDelayHDX in line 14 is equal to those produced by the two other approaches. The computational experiments ran as a single thread on a 2.40 GHz Intel Quad Core processor with 8 GB of RAM, running 64 bit Linux.

Table I reports on the total time (in seconds) required to find all solutions on the four proteins Cabin, CytoC, FKBP, and Myoglobin of horse heart. The biochemical experiments were performed at the National High Magnetic Field Laboratory in Tallahassee, Forida. The aggregate exchange rates for the digested fragments of the proteins were computed from the deuterium uptake by a novel ILP approach [Althaus et al.

[3] http://www.cplex.com

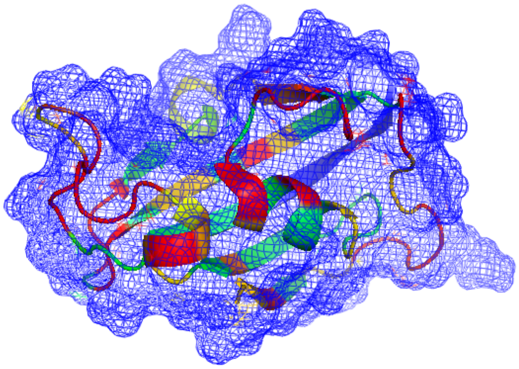


Fig. 5. Protein FKBP with its residues colored according to a prediction of rate kinetics obtained by averaging over approximate solutions. Color red encodes a high exchange rate, color orange denotes a medium rate, and residues colored green are predicted to exchange at low rate.

2009], except for FKBP V2, which was analyzed by the MEM method [Zhang et al. 1997]. FKBP was digested either by pepsin (FKBP V1 and FKBP V2) or by xiii (FKBP V3), FKBP V4 combines both datasets. Note that we can decompose instances into independent subproblems $(A, B, \ldots)$ if fragments from different subproblems do not overlap.

PolyDelayHDX is much more efficient than the Lagrangian method, and it performs slightly better than the ILP approach on most of the instances. However, on the most difficult instance (FKBP V2) we perform worse than the ILP approach. The main reason for the overall similar performance of the ILP approach of Althaus *et al.* and our enumeration algorithm is that the real world instances (available to us) were of small size to the extent that all solutions turned out to be integral; hence the ILP behaves essentially the same as an LP, and we do not have to apply our rounding procedure. Therefore, for instance FKBP V2 more elaborate branching strategies used by state-of-the-art ILP solvers have a larger impact on the overall performance.

Figure 5 gives a structural view of our results on protein FKBP. As motivated in the introduction, buried parts of the protein show on average lower exchange rates than parts that are exposed to the solvent.

We also generated a number of synthetic instances to study the performance of our algorithm on larger instances. While an increase of n is obviously justified by the existence of proteins with more than 152 residues (the largest real world instance considered by Althaus et al. [2010]), a higher number of colors will allow us to differentiate between the exchange kinetics of residues at a higher resolution. Each instance is the result of the following procedure, which starts with a real protein whose structure is already known. For each protein we computed the solvent accessibility of each residue using Naccess [Hubbard and Thornton 1993]. The idea behind this is that residues that are more accessible will show a higher exchange rate; thus, solvent accessibility is a good predictor of exchange rate. To model the noisy nature of real experiments, we added a small Gaussian random variable to these values. The perturbed solvent accessibility was then discretized uniformly into as many colors as available. In the following discussion, we refer to this coloring as the *true coloring*. Finally, our instance was created by choosing randomly chosen intervals and aggregating the information

Table I. Comparison of running times in seconds of the ILP based branching approach by Althaus et al. [2010] (ILP-Branching), the Lagrangian relaxation approach [Canzar 2008] (Lagrange) and our polynomial-delay enumeration algorithm (PolyDelayHDX), ran in type 1 mode ($h = 0$), on proteins Cabin, CytoC, FKBP, and Myoglobin of horse heart. The instances are characterized by the number of residues n, the number of intervals (fragments) $|\mathcal{I}|$, the number of segments $|\mathcal{S}|$, and the minimal total error ϵ. The number of colors in these instances is 3, representing low, medium, and high exchange rates. All three approaches are based on the same ILP formulation and hence produce the same number of solutions $\#Sol$.

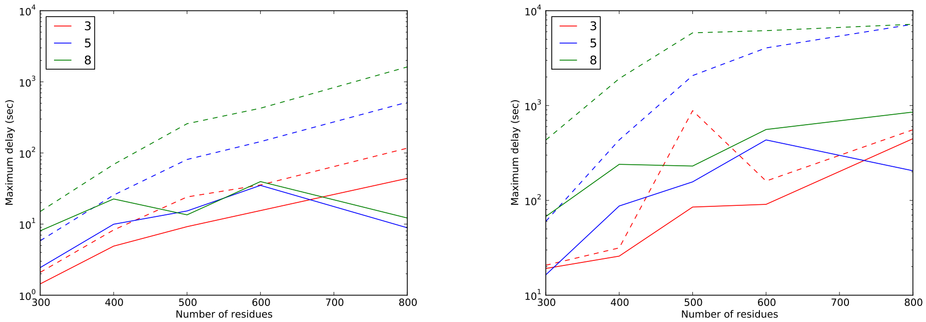
Instance	n	$\|\mathcal{S}\|$	$\|\mathcal{I}\|$	ϵ	#Sol	ILP-Branching	Lagrange	PolyDelayHDX
Cabin	78	26	34	128	36	3.82	7.93	0.87
CytoC	74	18	17	40	1980	0.31	5.89	0.16
CytoC-A	27	5	6	6	1	0.02	0.01	0.01
CytoC-B	26	5	6	30	110	0.25	5.63	0.12
CytoC-C	15	6	5	4	18	0.04	0.25	0.03
FKBP V1 (ilp)	101	34	31	47	37800	1.32	128.97	1.37
FKBP V1-A	35	15	12	15	126	0.75	30.22	0.90
FKBP V1-B	16	5	5	4	4	0.02	0.06	0.01
FKBP V1-C	36	12	14	28	75	0.55	98.69	0.46
FKBP V2 (mem)	101	34	31	46	1160040	11.31	523.84	33.92
FKBP V2-A	35	15	12	16	840	2.7	285.94	7.43
FKBP V2-B	16	5	5	2	1	0.01	0.02	0.01
FKBP V2-C	36	12	14	28	1381	8.6	237.88	26.48
FKBP V3 (xiii)	103	34	47	38	6	0.17	0.13	0.08
FKBP V3-A	22	10	16	12	1	0.03	0.01	0.02
FKBP V3-B	10	4	4	2	3	0.02	0.02	0.01
FKBP V3-C	11	5	4	0	1	0.01	0.03	0.01
FKBP V3-D	25	10	22	24	2	0.10	0.06	0.03
FKBP V3-E	3	1	1	0	1	0.01	0.01	0.01
FKBP V4 (both)	105	43	56	58	1536	1.02	6.54	0.50
FKBP V4-A	49	20	24	18	24	0.54	5.39	0.32
FKBP V4-B	11	5	4	0	2	0.01	0.01	0.01
FKBP V4-C	25	12	26	40	16	0.46	1.12	0.16
FKBP V4-D	4	3	2	0	2	0.01	0.02	0.01
HorseHeart	152	49	48	42	1121760	0.82	11.95	0.56
HorseHeart-A	17	9	10	14	20	0.13	2.05	0.05
HorseHeart-B	12	2	4	2	2	0.02	0.01	0.01
HorseHeart-C	22	8	8	8	82	0.22	7.42	0.21
HorseHeart-D	37	14	17	14	38	0.37	2.31	0.24
HorseHeart-E	3	1	1	0	1	0.01	0.01	0.01
HorseHeart-F	21	6	6	4	9	0.04	0.12	0.02
HorseHeart-G	4	1	1	0	1	0.01	0.01	0.01
HorseHeart-H	7	1	1	0	1	0.02	0.02	0.01

from the true coloring. The number of intervals was chosen to be half the length of the protein, following the distribution we found on real instances. Our testbed was made up of 22 proteins chosen from the Protein Data Bank [Berman et al. 2000]. The proteins ranged in length from 250 to 750 residues each. For each protein we created 3 instances, distinguishing between 3, 5, and 8 different exchange rates.

We ran the experiments on the synthetic instances single threaded on a compute cluster with two 2.83GHz Intel Xeon CPU's and 16 Gbytes of RAM on each node, running the Scientific Linux 5.4 operating system. Computations exceeding a time limit of 2 hours were aborted. In what follows, we refer to solutions output in line 24 as rounded solutions (colorings), where the approximation always is with respect to an optimal (fractional) solution to the LP relaxation computed in the root node. In contrast, integral solutions (colorings) refer to solution output in the leaves of the search tree in line 14. However, if no such integral solution can be found due to a non-integral reference solution computed in the root node, combined with a small value of h, the approximation of integral solutions will be with respect to the optimal solution to the root ILP (instead of the LP relaxation). For type 1 and $h = 0$ these solutions correspond one-to-one to the solutions output by the 2 alternative branching approaches [Althaus et al. 2010] and [Canzar 2008]. An optimal (integral) solution always refers to a solution output in line 14 for $h = 0$ or the reference solution computed in the root node. Where it is not clear from the context, we explicitly point out which case applies. In the experiments described in the following we investigate the impact rounded colorings have on different aspects of a branching-based enumeration method. For that, we compare PolyDelayHDX that produces both integral (as defined above) and rounded colorings to a version that outputs exclusively integral solutions. We commonly refer to the former set of solutions as approximate solutions. We also run PolyDelayHDX for fractional values of h to allow for a larger set of rounded solutions.

Figure 6 shows the effectiveness of the proposed method in producing colorings at much higher rate than those that can be produced by exact integer programming solvers. In the case of 3 different exchange rates the delay increases with an increasing number of residues, but still allows to derive a new approximate coloring of type 0 within seconds (Figure 6(a)). For 5 and 8 exchange rates the delay remains rather stable along the considered range of proteins sizes. Furthermore, the figure shows that especially for larger proteins and a higher number of exchange rates, the maximum delay between two consecutive 2-approximate colorings of type 0 is considerably smaller than the maximum delay between two integral solutions. While rounded colorings reduce the maximum delay in the case of 3 exchange rates and proteins of size 800 residues on average by a factor of $\sim$3, they reduce the maximum delay for 5 and 8 exchange rates on average by a factor of $\sim$14, respectively $\sim$100. A simple explanation for this observation is the increased rate of rounded solutions for larger proteins and a higher number of exchange rates (see Figure 7(a)).

Concerning the delay between 2-approximate colorings of type 1 (Figure 6(b)), instances for which no further integral solution was encountered within the 2 h time limit contributed 7200 seconds to the values captures by the dashed lines. This might lead to an underestimation of the delay between integral solutions, especially in the case of 5 and 8 different exchange rates, for proteins larger than 500, respectively 600 residues. If only 3 exchange rates are considered, the maximum delay behaves similarly, independent of whether rounded solutions are output. Notable exceptions are two proteins comprised of 500 residues, for which not a single integral solution was found in the 2 h time limit. In contrast, on one of these two proteins PolyDelayHDX evaluated the complete branching tree within 5 seconds, for the other protein it provided rounded solutions with a maximum delay of 36 seconds. For a higher number of exchange rates the rounded solutions reduce the maximum delay considerably for



(a) 2-approximate colorings of type 0 produced by PolyDelayHDX

(b) 2-approximate colorings of type 1 produced by PolyDelayHDX

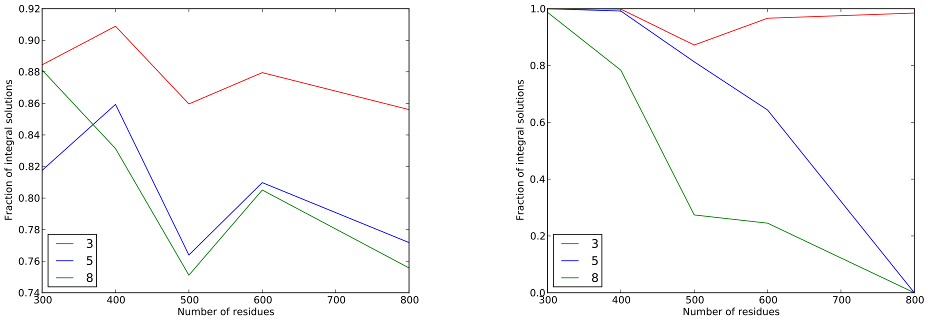
Fig. 6. Semi logarithmic plot of the maximum delay in seconds between two consecutive 2-approximate colorings produced by PolyDelayHDX (solid line) and between two optimal integral solutions (dashed line), distinguishing between 3, 5, and 8 different exchange rates. The values shown are obtained by averaging over the proteins contained in bins of size 50 residues.

type 1 approximations, especially when taking into account the upper bound of 7200 seconds on the maximum delay if no integral solution was found (see above). Again, this behaviour can be explained by the extremely low percentage of rounded solutions in the case of 3 exchange rates, and their rapidly increasing percentage for 5 and 8 exchange rates (see Figure 7(b)).

In Figure 8 we examine the effect of the error bound h on the maximum delay between two consecutive approximate colorings of type 0. It is worth noting that both our definition of approximate coloring and the guarantee of Theorem 4.2 only hold for integral h. However, the algorithm is well defined for fractional h so in our experiments we explore its behavior for factional parameters of h as well. As the set of integral solutions output by our algorithm in line 14 is identical for all $h < 1$, provided the reference solution computed in the root node is integral, the delay between optimal integral solutions is measured, for all $h < 1$, by running PolyDelayHDX with h set to 0. By that we avoid a negative impact of an increased number of rounded solutions on the delay between integral solutions. Since the maximum delay is of greater importance for larger proteins (see Figure 6), Figure 8 focuses on proteins of size at least 600 residues. The figure shows that the influence of parameter h on the maximum delay is rather limited for $h < 1$. After an initial increase for $h \leq 0.1$ (3 exchange rates, see Figure 8(a)), respectively $h \leq 0.2$ (5 exchange rates, see Figure 8(b)), the maximum delay shows now clear tendency for $h < 1$. For 3 exchange rates (Figure 8(a)) the maximum delay between optimal integral and approximate colorings decreases, respectively increases for $h = 1$. For 5 exchange rates (Figure 8(b)) the maximum delay of both approaches increases for $h = 1$. In general, for 3, 5, and 8 exchange rates PolyDelayHDX becomes less effective for $h = 1$, as significantly less colorings are obtained by rounding (see Figure 9).

To evaluate the biochemical relevance of colorings output by PolyDelayHDX we compare the distortions of approximate colorings with respect to the known true coloring of the synthetic instances to the distortions of the presumably optimal (for $h = 0$) integral solutions.

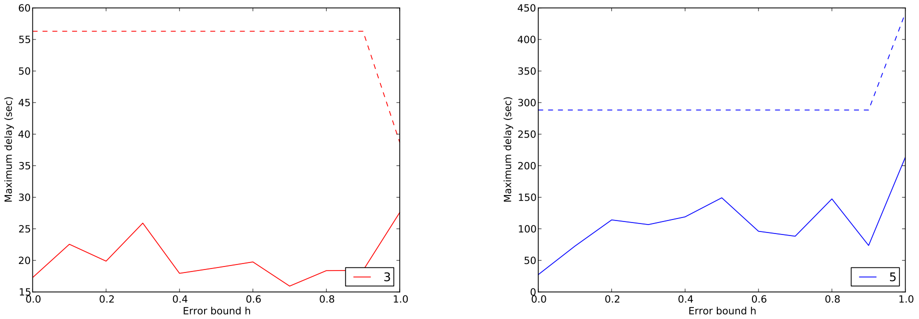
Since we know the true coloring of a synthetic instance we can compute the distortion at the highest possible resolution, namely on the level of segments, rather than on the level of fragments. The distortion of a coloring χ output by our algorithm in line 14 or



(a) 2-approximate colorings of type 0 produced by PolyDelayHDX

(b) 2-approximate colorings of type 1 produced by PolyDelayHDX

Fig. 7. Percentage of integral colorings that were not obtained by rounding, distinguishing betweeen 3, 5, and 8 different exchange rates. The values shown are obtained by averaging over the proteins contained in bins of size 50 residues.



(a) 3 exchange rates

(b) 5 exchange rates

Fig. 8. Maximum delay in seconds between consecutive outputs. The error bound h varies in the range $[0, 1]$ and we search for approximate colorings of type 0. The output of PolyDelayHDX is shown as a solid line and the optimal integral colorings, as a dashed line. Considering proteins of size at least 600 residues. The set of optimal integral solutions does not change for $h < 1$.

line 24 with respect to a true coloring χ^* is then given by:

$$\sum_{s \in \mathcal{S},\ i \in [k]} |\chi(s,i) - \chi^*(s,i)|.$$

Figure 10 compares the distortion of 2-approximate colorings of both types output by PolyDelayHDX to the distortion of integral solutions. In the case of type 1 approximations, the optimal ($h = 0$) integral solutions are exactly the solutions output by the approach of Althaus et al. [2010]. While the total distortion increases with the size of the proteins, the difference in distortion remains marginal, especially in the case of 3 and 5 different exchange rates. For proteins of up to ∼500 residues in size the distortion of 2-approximate colorings produced by PolyDelayHDX for 3 different exchange rates is slightly smaller than the distortions of optimal integral colorings. For larger proteins the 2-approximate colorings of types 0 and type 1 have a distortion that is at most ∼1%, respectively at most ∼2% larger than the distortion of the optimal integral

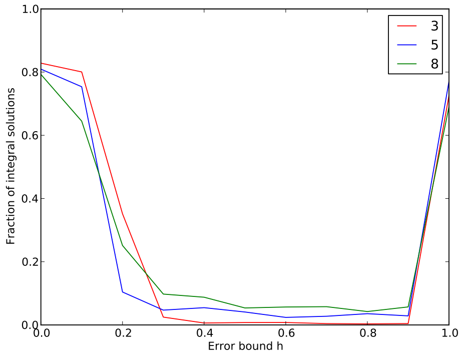
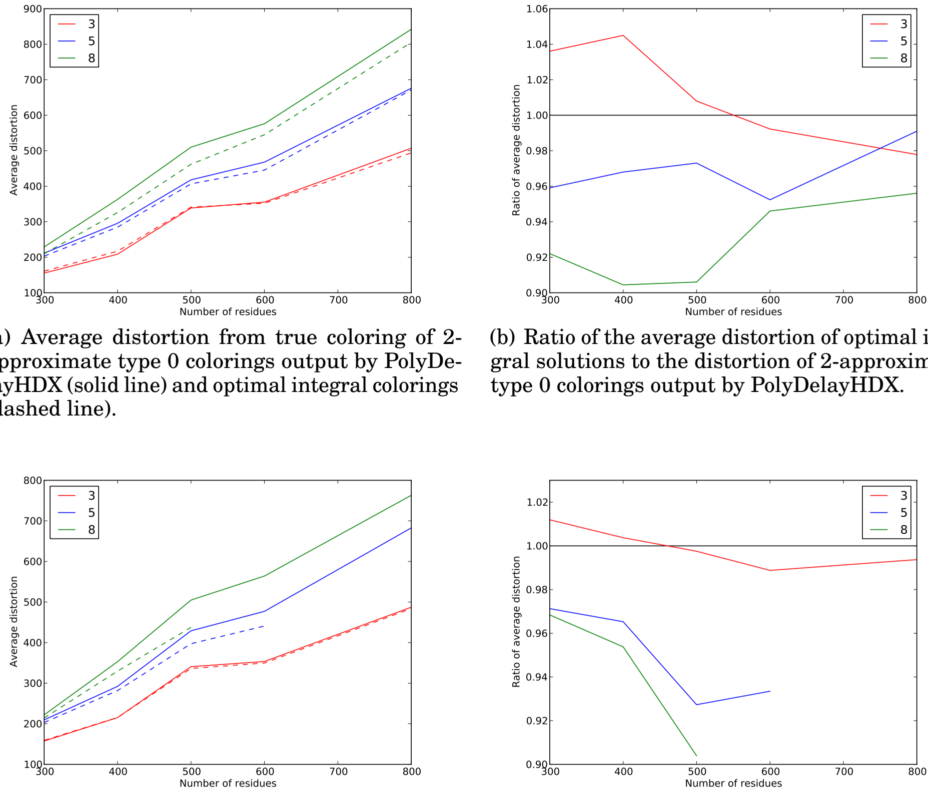


Fig. 9. Percentage of integral colorings relative to the total number of output solutions, distinguishing between 3, 5, and 8 different exchange rates. Only proteins of size at least 600 residues were taken into account.

colorings. The difference in distortion increases with a higher number of exchange rates. Nevertheless, for 5 and 8 exchange rates the distortion gap of type 0 approximations decreases from ∼5%, respectively ∼10%, to ∼1% respectively ∼4% for larger proteins. The quality of 2-approximate type 1 colorings seems to depend to a higher degree on the number of exchange rates than type 0 approximate colorings. However, in the case of 5 and 8 exchange rates only for proteins of size up to 600, respectively 500 residues, an additional optimal integral solution could be found during the implicit enumeration within the allowed time limit.

In Figure 11 we study how close the best among all output colorings is to the known true coloring. Following the intuition that the true coloring might not satisfy all constraints imposed by the reference solution computed in the root node, we plot the minimal distortion of all solutions of type 0 output for error bound $h \in \{0, 0.1, 0.2, \ldots, 1\}$. Note that the set of integral solutions and thus their minimal distortion does not change for $h < 1$. For 3 and 5 exhange rates, increasing h to $\sim 0.3 - 0.4$ allows for approximate solutions that are closer to the true coloring than the integral solution by ∼14%, respectively ∼5%. Although the distortion decreases for 8 exchange rates when increasing h from 0 to ∼0.3, the quality of the approximate solutions is worse than the quality of the optimal integral solutions for all $h \leq 0.9$. For $h = 1$, when the set of integral solutions changes, the approximate solutions have a 5% smaller distortion than the integral solutions found in the 2 hour time limit. In general, for all considered exchange rates and both integral and approximate colorings, the minimim distortion increases for $h = 1$. This is mostly due to the enormous increase in size of the set of integral solutions, which cannot be entirely explored in the given time limit.

We plot in Figure 9 the percentage of integral solutions that were not obtained by rounding as a function of h, for $0 \leq h \leq 1$. Confirming our previous observations concerning the minimal distortion (Figure 11) and the maximum delay (Figure 6), the effectiveness of our approach increases rapidly already for small h. While for $h = 0$ only around 20% of the enumerated colorings are obtained through rounding in line 24, for $h \approx 0.15$ roughly every second solution results from our rounding procedure, and for $h > 0.3$ between 90% and 98%, depending on the number of exchange rates considered, are output in line 24. For $h = 1$ the set of integral solutions increases drastically to a size, which does not allow for a single instance to explore the entire branching tree within the given time limit.



(a) Average distortion from true coloring of 2-approximate type 0 colorings output by PolyDelayHDX (solid line) and optimal integral colorings (dashed line).

(b) Ratio of the average distortion of optimal integral solutions to the distortion of 2-approximate type 0 colorings output by PolyDelayHDX.

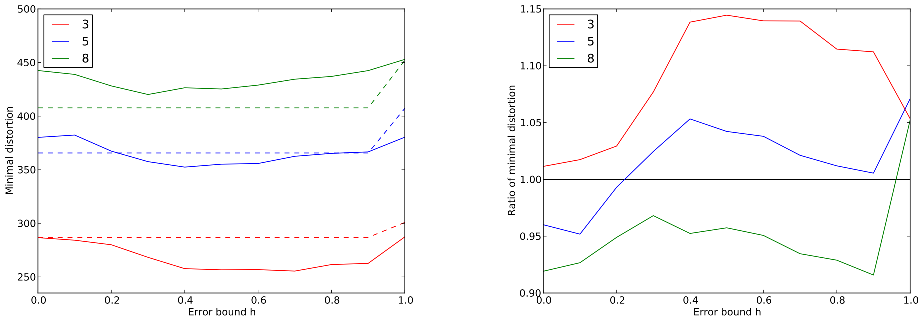
(c) Average distortion from true coloring of 2-approximate type 1 colorings output by PolyDelayHDX (solid line) and solutions produced by [Althaus et al. 2010] (dashed line).

(d) Ratio of the average distortion of optimal integral solutions produced by [Althaus et al. 2010] to the distortion of 2-approximate type 1 colorings output by PolyDelayHDX.

Fig. 10. Average distortion from true coloring of 2-approximate colorings, distinguishing between 3, 5, and 8 different exchange rates. The values shown are obtained by averaging over the proteins contained in bins of size 50 residues. *Upper row:* Type 0 approximation. *Lower row:* Type 1 approximation. In the case of 5 and 8 exchange rates, for proteins of size larger than 600, repsepcitvely 500 residues, no further integral solution was found through branching within the allowed time limit.

6. CONCLUSION

A new method for enumerating approximate solutions for the interval constrained coloring problem was evaluated. The proved theoretical guarantees, in terms of a low polynomial bound on the delay between successive solutions, were confirmed experimentally. In particular, comparing this method to the methods proposed earlier indicates that the new method is superior in the sense that the solutions are produced at much higher rate, while deviating only by slight margins from the optimal integral solutions. Even in the case of 3 exchange rates, where we only achieve a modest reduction in delay on average, our method does not allow for any outliers with extremely high delays (> 2 hours), which is in contrast to methods that are based solely on integral solutions that are not obtained by rounding (see Section 5), like the branching method by Althaus et al. [2010]. Furthermore, for 3 and 5 exchange rates the best solution found is almost always closer to the true coloring if rounded solutions are generated. Moreover,



(a) Minimal distortion from true coloring of solutions output by PolyDelayHDX (solid line) and optimal integral solutions (dashed line).

(b) Ratio of the minimal distortion of optimal integral solutions to the distortion of colorings output by PolyDelayHDX.

Fig. 11. Minimal distortion from true coloring of approximate colorings of type 0. The error bound h varies in the range $[0, 1]$, and we distinguish between 3, 5, and 8 different exchange rates.

the number of solutions and the quality of approximation can be somewhat controlled by setting a single parameter h, which gives the biochemist enough flexibility in controlling the trade off between producing a smaller number of solutions and reducing the delay between the different solutions. Such techniques might prove useful for other problems of similar nature.

Clearly, the applicability of the distortion from a “true” solution as a measure of solution quality inherently depends on the extent to which our random error model captures the real measurement errors.

Our implementation of the polynomial delay enumeration algorithm, PolyDelayHDX, as well as the synthetic instances used in this paper are freely available at http://ccb.jhu.edu/people/canzar/software.html.

ACKNOWLEDGMENT

We thank Anke Meyer-Bäse and Ernst Althaus for providing the real-world instances for our experiments.

REFERENCES

Ernst Althaus, Stefan Canzar, Carsten Ehrler, Mark Emmett, Andreas Karrenbauer, Alan Marshall, Anke Meyer-Base, Jeremiah Tipton, and Hui-Min Zhang. 2010. Computing H/D-Exchange rates of single residues from data of proteolytic fragments. *BMC Bioinformatics* 11, 1 (2010), 424. DOI:http://dx.doi.org/10.1186/1471-2105-11-424

Ernst Althaus, Stefan Canzar, Carsten Ehrler, Mark R. Emmett, Andreas Karrenbauer, Alan G. Marshall, Anke Meyer-Bäse, Jeremiah Tipton, and Huimin Zhang. 2009. Discrete Fitting of Hydrogen-Deuterium-exchange-data of overlapping fragments. In *Proceedings of the 4th International Conference on Bioinformatics & Computational Biology*. 23–30.

Ernst Althaus, Stefan Canzar, Khaled M. Elbassioni, Andreas Karrenbauer, and Julián Mestre. 2011. Approximation Algorithms for the Interval Constrained Coloring Problem. *Algorithmica* 61, 2 (2011), 342–361.

Ernst Althaus, Stefan Canzar, Mark R. Emmett, Andreas Karrenbauer, Alan G. Marshall, Anke Meyer-Bäse, and Huimin Zhang. 2008. Computing H/D-exchange speeds of single residues from data of peptic fragments. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing*. 1273–1277.

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. 2000. The Protein Data Bank. *Nucleic Acids Res* 28 (2000), 235–242.

M. R. Bussieck and M. E. Lübbecke. 1998. The vertex set of a 0/1 polytope is strongly $\mathcal{P}$-enumerable. *Computational Geometry* 11, 2 (1998), 103–109.

Jarek Byrka, Andreas Karrenbauer, and Laura Sanità. 2010. The Interval Constrained 3-Coloring Problem. In *Proceedings of the 9th Latin American Theoretical Informatics Symposium*. 591–602.

S. Canzar. 2008. *Lagrangian Relaxation - Solving NP-hard problems in Computational Biology via Combinatorial Optimization*. Ph.D. Dissertation. Universität des Saarlandes.

Stefan Canzar, Khaled M. Elbassioni, and Julián Mestre. 2010. A Polynomial Delay Algorithm for Enumerating Approximate Solutions to the Interval Constrained Coloring Problem. In *ALENEX*. 23–33.

S. W. Englander. 2006. Hydrogen exchange and mass spectrometry: A historical perspective. *Journal of the American Society for Mass Spectrometry* 17, 11 (2006), 1481–1489. DOI:http://dx.doi.org/10.1016/j.jasms.2006.06.006

Fred Glover. 1967. Maximum matching in a convex bipartite graph. *Naval Research Logistics Quarterly* 14, 3 (1967), 313–316.

S.J. Hubbard and J.M. Thornton. 1993. NACCESS. (1993). http://www.bioinf.manchester.ac.uk/naccess/.

Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. 2011. Deconstructing intractability - A multivariate complexity analysis of interval constrained coloring. *J. Discrete Algorithms* 9, 1 (2011), 137–151.

J.F. Leite and M. Cascio. 2002. Probing the Topology of the Glycine Receptor by Chemical Modification Coupled to Mass Spectrometry. *Biochemistry* 41, 19 (2002), 6140–6148. http://pubs3.acs.org/acs/journals/doilookup?in_doi=10.1021/bi015895m

A. Schrijver. 2003. *Combinatorial Optimization: Polyhedra and Efficiency, Algorithms and Combinatorics*. Vol. 24. Springer, New York.

J.S. Sharp, J.M. Becker, and R.L. Hettich. 2004. Analysis of Protein Solvent Accessible Surfaces by Photochemical Oxidation and Mass Spectrometry. *Analytical Chemistry* 76, 3 (2004), 672–683. http://pubs3.acs.org/acs/journals/doilookup?in_doi=10.1021/ac0302004

Z. Zhang, W. Li, T. M. Logan, M. Li, and A. G. Marshall. 1997. Human recombinant [C22A] FK506-binding protein amide hydrogen exchange rates from mass spectrometry match and extend those from NMR. *Protein Science* 6, 10 (1997), 2203–2217.

E.R.P. Zuiderweg. 2002. Mapping Protein-Protein Interactions in Solution by NMR Spectroscopy. *Biochemistry* 41, 1 (2002), 1–7. http://pubs3.acs.org/acs/journals/doilookup?in_doi=10.1021/bi011870b