Theoretical Computer Science 593 (2015) 42–50

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The interval constrained 3-coloring problem

Jaroslaw Byrka [a,*], Andreas Karrenbauer [b], Laura Sanità [c]

[a] Institute of Computer Science, University of Wroclaw, Poland
[b] Max Planck Institute for Informatics, Saarbrücken, Germany
[c] Combinatorics and Optimization Department, University of Waterloo, Canada

ARTICLE INFO

Article history:
Received 9 October 2012
Received in revised form 19 April 2015
Accepted 26 April 2015
Available online 3 June 2015
Communicated by V.Th. Paschos

Keywords:
Interval constrained coloring
APX-hardness

ABSTRACT

In this paper, we settle the open complexity status of interval constrained coloring with a fixed number of colors. We prove that the problem is already NP-complete if the number of different colors is 3. Previously, it has only been known that it is NP-complete, if the number of colors is part of the input and that the problem is solvable in polynomial time, if the number of colors is at most 2. We also show that it is hard to satisfy almost all of the constraints for a feasible instance (even in the restricted case where each interval is used at most once). This implies APX-hardness of maximizing the number of simultaneously satisfiable intervals.

1. Introduction

In the interval constrained 3-coloring problem, we are given a set $\mathcal{I}$ of intervals defined on $[n] := \{1, \dots, n\}$ and a *requirement* function $r : \mathcal{I} \to \mathbb{Z}^3_{\geq 0}$, which maps each interval to a triple of non-negative integers. The objective is to determine a coloring $\chi : [n] \to \{1, 2, 3\}$ such that each interval gets the proper colors as specified by the requirements, i.e., $\sum_{i \in I} e_{\chi(i)} = r(I)$ where e_1, e_2, e_3 are the three unit vectors of $\mathbb{Z}^3$.

This problem is motivated by an application in biochemistry to investigate the tertiary structure of proteins as shown in the illustration below (Fig. 1). More precisely, in Hydrogen-Deuterium-Exchange (HDX) experiments proteins are put into a solvent of heavy water (D_2O) for a certain time after which the amount of residual hydrogen atoms, that have exchanged with deuterium atoms, is measured [1]. Doing this experiment for several timesteps, one can determine the exchange rate of the residues. These exchange rates indicate the solvent accessibility of the residues and hence they provide information about the spatial structure of the protein. Mass spectroscopy is one of the methods for measuring these exchange rates. To this end, the proteins are digested, i.e., cut into parts which can be considered as intervals of the protein chain, and the mass uptake of each interval is measured. But thereby only bulk information about each interval can be obtained. Since there is not only one protein in the solvent but millions and they are not always cut in the same manner, we have this bulk information on overlapping fragments. That is, we are given the number of slow, medium, and fast exchanging residues for each of these intervals and our goal is to find a feasible assignment of these three exchange rates to residues such that for each interval the numbers match with the bulk information. See [2] for more discussion on the relevance of the three exchange rates setting.

Though the interval constrained 3-coloring problem is motivated by a particular application, its mathematical abstraction appears quite simple and ostensibly more general. In terms of integer linear programming, the problem can be equivalently

* Corresponding author.
E-mail addresses: jby@ii.uni.wroc.pl (J. Byrka), andreas.karrenbauer@mpi-inf.mpg.de (A. Karrenbauer), lsanita@uwaterloo.ca (L. Sanità).

http://dx.doi.org/10.1016/j.tcs.2015.04.037
0304-3975/

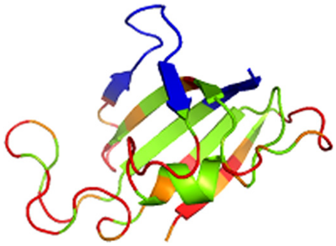

Fig. 1. Coloring of the residues of a protein chain according to their exchange rates.

formulated as follows. Given a matrix $A \in \{0,1\}^{m\times n}$ with the *row-wise consecutive-ones property* and three vectors $b_{1,2,3} \in \mathbb{Z}^m_{\geq 0}$, the constraints

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \\ I & I & I \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ 1 \end{pmatrix} \tag{1}$$

have a binary solution, i.e. $x_{1,2,3} \in \{0,1\}^n$, if and only if the corresponding interval constrained 3-coloring problem has a feasible solution. We may assume w.l.o.g. that the requirements are consistent with the interval lengths, i.e. $A \cdot \mathbb{1} = b_1 + b_2 + b_3$, since otherwise we can easily reject the instance as infeasible. Hence, we could treat x_3 as slack variables and reformulate the constraints as

$$Ax_1 = b_1, \qquad Ax_2 = b_2, \qquad x_1 + x_2 \leq 1. \tag{2}$$

It is known that if the matrix A has the *column-wise* consecutive ones property (instead of *row-wise*), then there is a reduction from the two-commodity integral flow problem, which has been proven to be NP-complete in [3]. However, the NP-completeness w.r.t. row-wise consecutive ones matrices has been an open problem in a series of papers as outlined in the following subsection.

1.1. Related work

The problem of assigning exchange rates to single residues has first been considered in [4]. In that paper, the authors presented a branch-and-bound framework for solving the corresponding coloring problem with k color classes. They showed that there is a combinatorial polynomial time algorithm for the case of $k = 2$. Moreover, they asked the question about the complexity for $k > 2$. In [5] (see also the extended version [6]) the problem has been called *interval constrained coloring*. It has been shown that the problem is NP-hard if the parameter k is part of the input. Moreover, approximation algorithms have been presented that allow violations of the requirements. That is, a quasi-polynomial time algorithm that computes a solution in which all constraints are $(1+\varepsilon)$-satisfied and a polynomial time rounding scheme, which satisfies every requirement within ± 1, based on a technique introduced in [7]. The latter implies that if the LP relaxation of (1) is feasible, then there is a coloring satisfying at least $\frac{5}{16}$ of the requirements. APX-hardness of finding the maximum number of simultaneously satisfiable intervals has been shown in [8] for $k \geq 2$ provided that intervals may be counted with multiplicities. But still, the question about the complexity of the decision problem for fixed $k \geq 3$ has been left open. In [9], several fixed parameter tractability results have been given. However, the authors state that they do not know whether the problem is tractable for fixed k.

Recently Canzar et al. have studied the problem of enumerating close to optimal solutions to interval constrained coloring. In [10] they provide a polynomial-delay algorithm which frequently outputs solutions that are close to optimal and potentially valuable in practice.

1.2. Our contribution

In this paper, we prove the hardness of the interval constrained k-coloring problem for fixed parameter k. In fact, we completely settle the complexity status of the problem, since we show that already the interval constrained 3-coloring problem is NP-hard by a reduction from 3-SAT. This hardness result holds more generally for any problem that can be formulated like (1). Moreover, we show an even stronger result: it is still difficult to satisfy almost all of the constraints for a feasible instance. More precisely, we prove that there is a constant $\epsilon > 0$ such that it is NP-hard to distinguish between instances where all constraints can be satisfied and those where only a $(1-\epsilon)$ fraction of constraints can be simultaneously satisfied. To this end, we extend our reduction using expander graphs. This gap hardness result implies APX-hardness of the problem of maximizing the number of satisfied constraints for $k \geq 3$. It is important to note that our construction does neither rely on multiple copies of intervals nor on inconsistent requirements for an interval, i.e., in our construction for every interval (i, j) we have unique requirements that sum up to the length of the interval.