ARTICLE IN PRESS
JDA:660

Journal of Discrete Algorithms ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Reconstructing binary matrices with timetabling constraints

Srecko Brlek [a], Stefano Brocchi [b], Andrea Frosini [b,*]

[a] Laboratoire de combinatoire et d'informatique mathématique, UQAM, Canada
[b] Dipartimento di Matematica e Informatica "Ulisse Dini", Università degli Studi di Firenze, Viale Morgagni 65, 53134, Firenze, Italy

ARTICLE INFO

Article history:
Received 20 June 2013
Received in revised form 3 May 2016
Accepted 3 July 2016
Available online xxxx

Keywords:
Discrete tomography
Reconstruction problem
Computational complexity
Scheduling problem

ABSTRACT

This paper deals with the reconstruction of binary matrices having specific local constraints, called *Timetable Constraints*, imposed on their elements. In the first part of the paper, we show that instances of this problem with some fixed parameters are solvable in linear time by means of a greedy approach. In the following, thanks to a slight relaxation of one of the Timetable Constraints, we define a second and more complex greedy algorithm that solves a much wider family of instances. In both of these cases the computational complexity of our algorithms is linear in the dimension of the reconstructed matrix. Finally, we complete the framework by proving that all but one the remaining cases are NP-complete.

1. Introduction

Each time we try to allocate a finite number of resources over time in order to accomplish a set of tasks that may span from performing computations on a processor, to managing the production lines of a factory or, in an every-day life scenario arranging our busy Saturday evening, then we are facing a scheduling problem. A variety of these problems appear in real-world situations (see [2] for a recent survey and [3,5] for practical applications), and, as one can easily imagine, they do not allow a uniform approach. In fact, almost every problem exhibits specific characteristics, including scheduling constraints, and needs specific techniques borrowed from mathematics and computer science to be efficiently handled (see [6]).

At an organization level, scheduling constraints often limit the type of possible social structures, and they usually involve both the availability of structures over time, and the social constraints imposed by each employee's time slots. The related problems may vary into a huge range of different situations, including timetable designing, staff scheduling, production planning, etc.

Our study focuses on the *quick time* implementation of a working-timetable for employees sharing a limited and variable set of resources in the case where each of them has to perform a job requiring a determined number of time slots. We further assume that each job can be split into tasks, under specific social constraints, and we admit as acceptable solutions only those presenting an alternation of working and rest periods satisfying them; such a scenario fits in the paradigm of the *constrained preemptive scheduling* with dedicated agents. We refer to this problem as Timetable Problem (briefly *TP*).

Many other realistic scenarios share the same model. As an example, the working plan of some machines that can be active for a fixed number of consecutive time slots at most, and then they have to rest (say, to cool off) for a fixed

* Corresponding author.
E-mail address: andrea.frosini@unifi.it (A. Frosini).

http://dx.doi.org/10.1016/j.jda.2016.07.001
1570-8667/© 2016 Published by Elsevier B.V.

minimum number of time slots. This variety of possible situations makes our study relevant for application to many real world problems. Similar studies have been recently carried on using techniques that are mainly based on Graph Theory (see [11,12]) or Discrete Geometry (see [7,9,14]).

Scheduling problems belong to the wider class of combinatorial search problems, where each instance consists of a finite set of parameters that have to satisfy some constraints, and the solution is required to belong to a definite space. According to this latter, the problems divide into two classes: the optimization ones and the decision ones. The first class requires require to search in a (usually wide) space a solution that optimizes some objective function, while the other one requires a decision, i.e. a boolean solution set.

So, in general, scheduling problems are viewed as optimization problems where we try to minimize the span time of a given set of jobs performed by agents, sometimes in presence of limited availability of resources or constraints that act either on the jobs' precedence or on the jobs' mutual interaction.

A variety of techniques have been designed, mainly based on enumerative methods (for a basic overview see [23], for a comparative study of the methods see [10], while for details in case of limited resources see [19]), and approximation algorithms (see [18,24] for surveys, and [27] for interesting open problems), sometimes successfully inserted in an artificial intelligence paradigm.

In our approach, the standard scenario of a preemptive allocation of jobs with limited resources is combined with the specificity of social constraints that realistically limit the performable tasks by imposing geometrical properties on them. Such a mix enables us to address the *TP* with techniques developed specifically in the area of Discrete Tomography in order to reach a double aim: first we determine in polynomial time one of the best possible solutions for a subclass of instances of *TP*, i.e. one of those, if any, where the span time is minimum, then we detect other subclasses where an optimal solution can not be reached in polynomial time.

Our study builds on and extends the one performed in [9], and more recently in [7], where only a limited subclass of instances of *TP* has been considered.

We also integrate [20,21], where the authors focus their attention on a similar problem but in a different context: the reconstruction of bar packings from horizontal and vertical projections, when both the length of the bars and the distance between them may vary in a fixed range. The results they obtain are somehow similar, but a key difference is present in the two models: on each row of the binary matrix representing the packing, they allow sequences of 0's of unbounded lengths at starting end ending periods. Such a difference is crucial, as in our context it is exactly the assignment of elements in the last two columns of the solution matrix that allows to derive P-solvable instances from NP-solvable ones.

The paper is organized as follows: for the rest of this section we introduce the timetable problem $TP(m,n,k)$, and we give necessary conditions for a solution to exist; Section 2 is devoted to the study of some basic properties of the instances of $TP(m,n,2)$ in order to design a simple polynomial time algorithm for the subproblem $TP(2,n,2)$. In Section 3, we build on the previous algorithm as starting point for producing a more complex one in Section 3. In this section we relax one of the social constraints, and we provide a nice generalization of the above algorithm; we also show that, in this case, it is sufficient to set $k=2$ in order to further lower the computational complexity.

Finally, in the last section, we discuss the possibility of using this last algorithm to solve a subclass of instances of $TP(m,n,2)$, leaving open the problem of determining the computational complexity.

1.1. The basic model

Let us consider an $m\times n$ binary matrix $A=(a_{i,j})$, with $1\le i\le m$ and $1\le j\le n$. Its horizontal and vertical projections are respectively the vectors $\pi_h(A)=(h_1,\dots,h_m)$ and $\pi_v(A)=(v_1,\dots,v_n)$ of the sums of its elements for each row and each column, respectively, i.e.

$$h_i=\sum_{j=1\dots n} a_{i,j} \quad\text{and}\quad v_j=\sum_{i=1\dots m} a_{i,j}.$$

The matrix A is commonly used to model in a standard and natural way a huge number of different situations ranging from resource allocation to everyday life: among them, we focus on the preemptive scheduling of the tasks of m employees that span on n time slots (here *preemptive* meaning that each employee's task can be split into jobs under specific assumptions). More precisely, row i corresponds to the working plan of the ith employee, while column j corresponds to the jth time slot; the 0's are the working slots of the employees and the 1's are their slots-off. Each sequence of consecutive working slots is considered as a single job. Furthermore we impose some requirements to reflect social constraints that reflect in the elements of A

C1: $\forall i,j: 1\le i\le m, 1<j<n$, if $a_{i,j}=1$, then $a_{i,j-1}=1$ or $a_{i,j+1}=1$;
C2: for a fixed $k\in\mathbb{N}$, $k\ge 2$, $\forall i,j: 1\le i\le m, 1\le j'\le n-k+1$, we have

$$\sum_{l=0\dots k-1} a_{i,j'+l}>0.$$

Please cite this article in press as: S. Brlek et al., Reconstructing binary matrices with timetabling constraints, J. Discret. Algorithms (2016), http://dx.doi.org/10.1016/j.jda.2016.07.001