

Permuting matrices to avoid forbidden submatrices^{*}

Bettina Klinz^{*}, Rüdiger Rudolf, Gerhard J. Woeginger

^{*} *Technische Universität Graz, Institut für Mathematik B, Kopernikusgasse 24, A-8010 Graz, Austria*

Received 1 October 1992; revised 3 January 1994

Abstract

This paper attaches a frame to a natural class of combinatorial problems and points out that this class includes many important special cases.

A matrix M is said to avoid a set \mathcal{F} of matrices if M does not contain any element of \mathcal{F} as (ordered) submatrix. For \mathcal{F} a fixed set of matrices, we consider the problem of deciding whether the rows and columns of a matrix can be permuted in such a way that the resulting matrix M avoids all matrices in \mathcal{F} .

We survey several known and new results on the algorithmic complexity of this problem, mostly dealing with $\{0, 1\}$ -matrices. Among others, we will prove that the problem is polynomial time solvable for many sets \mathcal{F} containing a single, small matrix and we will exhibit some example sets \mathcal{F} for which the problem is NP-complete.

1. Introduction

Definitions

The *entries* of all matrices in this paper are nonnegative integers. For e some nonnegative integer, a $(0, e)$ -matrix is a matrix with entries from the set $\{0, 1, \dots, e\}$. By $r(M)$ and $c(M)$ we denote the number of rows and columns of matrix M , and by $f(M)$ we denote the number of non-zero entries in M . M^T is the transposed matrix of M . A matrix M contains a matrix M_1 as *submatrix*, if we can get M_1 by deleting a set of rows and columns from M . A matrix M *avoids* a set \mathcal{F} of matrices if no element of \mathcal{F} appears as submatrix in M . For \mathcal{F} a (finite or infinite) set of matrices, we denote by $AV(\mathcal{F})$ the set of all matrices that avoid \mathcal{F} .

Next, we define three sets of matrices corresponding to the three types of matrix permutations we will investigate.

^{*} This research has been supported by the Christian Doppler Laboratorium für Diskrete Optimierung and by the Fonds zur Förderung der wissenschaftlichen Forschung, Project P 8971-PHY.

^{*} Corresponding author.

- A matrix M is an element of the set $\mathcal{M}(\mathcal{F})$ if and only if there exists a permutation π_1 of the rows and a permutation π_2 of the columns such that the resulting matrix is in $\text{AV}(\mathcal{F})$.
- M is in the set $\mathcal{M}^s(\mathcal{F})$ if and only if it is a square matrix and if there exists a permutation π that when applied to rows and columns of M produces a matrix in $\text{AV}(\mathcal{F})$.
- Finally, M is in $\mathcal{M}^c(\mathcal{F})$ if and only if there exists a permutation π of the columns that leads to a member of $\text{AV}(\mathcal{F})$.

Now the following question arises.

- (Q) For some fixed set \mathcal{F} , is the membership problem in $\mathcal{M}(\mathcal{F})$ (or in $\mathcal{M}^s(\mathcal{F})$ or in $\mathcal{M}^c(\mathcal{F})$, respectively) polynomial time decidable?

We stress the fact that the set \mathcal{F} is *not* part of the input. It is clear that this membership problem is decidable in *exponential time* by simply checking all permutations of the rows and columns of some input matrix M . However, we will present sets \mathcal{F} for which the problem (Q) is NP-complete and other sets for which problem (Q) is polynomially solvable.

Alternative representations for (0, 1)-matrices

Let M be a given (0, 1)-matrix with $r(M)$ rows and $c(M)$ columns. There exist quite a number of ways to interpret M in graph-theoretical terms. In the sequel we will briefly mention just three important alternatives. (For a more complete treatise on this topic see [24, 10].)

- M can be represented by a bipartite graph $B(M) = (V_r, V_c; E)$ having a vertex in V_r for each row of M and a vertex in V_c for each column of M . There is an edge $(i, j) \in E$ joining vertices $i \in V_r$ and $j \in V_c$ if and only if the entry m_{ij} of M equals 1.
- If M is a square matrix it can be regarded as adjacency matrix of a directed graph $G(M) = (V, E)$ with an edge $(i, j) \in E$ if and only if $m_{ij} = 1$.
- One can associate a hypergraph $H(M) = (X, E_H)$ with M by viewing M as vertex-edge incidence matrix of H . There $X = \{x_1, \dots, x_{r(M)}\}$ denotes the set of vertices and $E_H = \{e_1, \dots, e_{c(M)}\}$ the set of (hyper)edges of the hypergraph H . The edges are defined by $x_i \in e_j$ if and only if $m_{ij} = 1$, $i = 1, \dots, r(M)$, $j = 1, \dots, c(M)$.

From the above it should be clear that for many sets \mathcal{F} of forbidden matrices there is a nice interpretation of the membership problems in $\mathcal{M}(\mathcal{F})$, $\mathcal{M}^s(\mathcal{F})$ or $\mathcal{M}^c(\mathcal{F})$ in terms of the graphs $G(M)$, $B(M)$ or the hypergraph $H(M)$, which can be associated with the input matrix M . Let us remark here that in the rest of the paper we will mainly concentrate on the matrix point of view and only sometimes mention alternative graph-theoretic interpretations explicitly.

Related results

Many results from graph and hypergraph theory concerning forbidden subgraphs are closely related to our problem (see e.g. [10]). Most of them, however, lead to sets \mathcal{F} of forbidden matrices and corresponding membership problems in $\mathcal{M}(\mathcal{F})$, $\mathcal{M}^s(\mathcal{F})$

or $\mathcal{M}^c(\mathcal{F})$ having the property that for each $F \in \mathcal{F}$ all permutations of F which are feasible for the membership problem under investigation are also in \mathcal{F} . (For $\mathcal{M}(\mathcal{F})$ all pairs of row and column permutations are feasible, for $\mathcal{M}^s(\mathcal{F})$ the same permutation has to be applied to rows and columns and for $\mathcal{M}^c(\mathcal{F})$ only permutations of the columns are feasible.) For the ease of exposition let us henceforth call sets \mathcal{F} of matrices with the symmetry property described above *highly symmetric*. For highly symmetric and finite \mathcal{F} , the membership problems in $\mathcal{M}(\mathcal{F})$, $\mathcal{M}^s(\mathcal{F})$ and $\mathcal{M}^c(\mathcal{F})$ degenerate to checking whether $M \in AV(\mathcal{F})$ (permutations of the input matrix M cannot help in avoiding \mathcal{F}).

Since in this paper the main focus lies on *permuting* rows and columns of a given matrix in order to avoid a set \mathcal{F} of forbidden submatrices, we will mainly concentrate on problems which can be formulated in terms of sets \mathcal{F} which are *not invariant* under row and column permutations. In Sections 2.1–2.9 we will review several examples of this type from the literature. There topics such as Γ -free or totally balanced matrices, greedy matrices, interval graphs, graph homomorphisms, (0, 1) Monge matrices and bipartite permutation graphs, which all are related to the subject of our paper, will be treated. For a short description of some applications where highly symmetric sets \mathcal{F} arise, e.g. balanced and strongly unimodular matrices, see Section 2.10.

Organization of the paper

Section 2 summarizes several known results that fit into our framework. In Section 3 we deal with the following equivalence problem: given two sets of forbidden submatrices, say \mathcal{F}_1 and \mathcal{F}_2 , we want to know whether they define the same set of matrices, i.e. whether or not $\mathcal{M}(\mathcal{F}_1) = \mathcal{M}(\mathcal{F}_2)$. This general question is rather difficult to answer, but we will give some non-trivial examples of equivalent sets \mathcal{F}_1 and \mathcal{F}_2 .

Section 4 investigates the complexity of recognizing membership in $\mathcal{M}(\mathcal{F})$ when \mathcal{F} contains a single small (1×3 or 2×2) matrix (Sections 4.1 and 4.2). Furthermore, some results for some special sets \mathcal{F} containing two or more 2×2 matrices are presented (Section 4.3). Section 5 contains a collection of NP-completeness results. Among others, we exhibit a set \mathcal{F} of two (0, 1)-matrices such that deciding membership in $\mathcal{M}(\mathcal{F})$ is NP-complete. We close the paper with a short discussion and some concluding remarks in Section 6.

2. Applications

In this section we summarize several problems and results from the literature that can be formulated as membership problems in some special sets $\mathcal{M}(\mathcal{F})$, $\mathcal{M}^s(\mathcal{F})$ and $\mathcal{M}^c(\mathcal{F})$.

2.1. Totally balanced matrices and gamma-free matrices

Let \mathcal{A}_k contain all $k \times k$ (0, 1)-matrices, $k \geq 3$, with no identical rows and columns and all of their row and column sums equal to 2. Matrices in \mathcal{A}_k are also called *cycle*

submatrices. A $(0, 1)$ -matrix M is called *totally balanced* (TB) if it does not contain a submatrix in \mathcal{A}_k for any $k \geq 3$, i.e. $M \in \text{AV}(\mathcal{F})$ with $\mathcal{F} = \bigcup_{k \geq 3} \mathcal{A}_k$ or equivalently $M \in \mathcal{M}(\mathcal{F})$. (In terms of the hypergraph $H(M)$ associated with M this means that each cycle of $H(M)$ of length ≥ 3 is required to have an edge containing three vertices of the cycle.)

Totally balanced matrices occur e.g. in location theory (see [10, 30, 35]). What makes TB matrices of particular interest to our paper is that apart from the highly symmetric characterization above, there is also a characterization in terms of a single forbidden submatrix. In the sequel we will describe this rather surprising relationship.

A $(0, 1)$ -matrix M is called Γ -free or row-inclusion matrix if it does not contain the submatrix

$$\Gamma = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

In other words, all of the rows i with $m_{ij} = 1$ are ordered by inclusion with respect to the columns $j, \dots, c(M)$. We then say that a $(0, 1)$ -matrix M has a Γ -free ordering if and only if $M \in \mathcal{M}(\{\Gamma\})$, i.e. there are row and column permutations π_1 and π_2 , respectively, such that the permuted matrix $M_{\pi_1, \pi_2} = (m_{\pi_1(i), \pi_2(j)})$ avoids the matrix Γ .

Now we are prepared to formulate the central theorem relating matrices in $\mathcal{M}(\{\Gamma\})$ and totally balanced matrices which has been obtained independently by several authors.

Theorem 2.1 (Anstee and Farber [5], Hoffman et al. [30] and Lubiw [34]). *A $(0, 1)$ -matrix M is totally balanced if and only if it has a Γ -free ordering, i.e. $M \in \mathcal{M}(\{\Gamma\})$.*

The best algorithm for deciding for a given input matrix M whether M lies in $\mathcal{M}(\{\Gamma\})$ is conceptually due to Lubiw [34] and was subsequently improved by Paige and Tarjan [38] by using clever data structures. We will sketch the main ideas of this approach in Section 4.2 when we shall discuss the membership problem in $\mathcal{M}(\mathcal{F})$ for sets \mathcal{F} consisting of a single 2×2 $(0, 1)$ -matrix.

2.2. Greedy matrices

Hoffman et al. [30] call a $(0, 1)$ -matrix *greedy* if it avoids the following two submatrices

$$G_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad G_2 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Let M be a $k \times n$ $(0, 1)$ greedy matrix, let $b \in \mathbb{R}_+^k$, $\bar{c} \in \mathbb{R}_+^n$ and $d \in \mathbb{R}_+^k$ be three nonnegative real vectors and let $\bar{c}_1 \geq \bar{c}_2 \geq \dots \geq \bar{c}_n \geq 0$. Then it can be shown (see [29, 30]) that successive maximization, i.e. a greedy approach, solves the linear program

$$\max \{ \bar{c}^T x \text{ s.t. } Mx \leq b, 0 \leq x \leq d \}.$$

This result motivates to ask for a characterization of all $(0, 1)$ -matrices which can be transformed into a greedy matrix by permuting rows and columns. In other words, we are interested into the set $\mathcal{M}(\{G_1, G_2\})$ where G_1 and G_2 are the two matrices defined above. The following rather surprising result states that a matrix can be transformed into a greedy matrix if and only if it has a Γ -free ordering.

Theorem 2.2 (Hoffman et al. [30]).

$$\mathcal{M}(\{G_1, G_2\}) = \mathcal{M}(\{\Gamma\}). \quad (1)$$

In [29] Hoffman tried to generalize the above results on greedy-solvability of linear program to arbitrary nonnegative matrices M . He introduced so-called *box-greedy* matrices which can again be defined by a set \mathcal{F} of forbidden submatrices. Unfortunately, for this set \mathcal{F} even the problem of deciding whether a matrix M lies in the set $\text{AV}(\mathcal{F})$ can be shown to be NP-complete (for further details see [29]).

2.3. Consecutive ones property, convex bipartite graphs and interval graphs

A $(0, 1)$ -matrix M has the *consecutive ones property for rows* if its columns can be permuted so that in each row all the ones are consecutive [21]. This means that a permutation of the columns is desired for which no two ones within a single row are separated by a zero in that same row. Equivalently, we may say that M is in $\mathcal{M}(\mathcal{F})$ or in $\mathcal{M}^c(\mathcal{F})$, for $\mathcal{F} = \{(1, 0, 1)\}$. Naturally, one can define the *consecutive ones property for columns* in an analogous way.

Booth and Lueker [11] showed how to recognize matrices with the consecutive ones property for rows in linear time by using so-called *PQ-trees* (cf. also Section 4.1 for a short description of the key idea of this type of trees).

Note that if a matrix M has the consecutive ones property for rows or for columns, then the bipartite graph $B(M)$ associated with M is *convex*. If M has both properties at the same time, $B(M)$ is even *doubly convex*. It is important to be able to recognize these graph classes, since the maximum cardinality matching problem is solvable in linear time for (doubly) convex graphs (see [33, 22]).

Furthermore, the consecutive ones property is closely related to *interval graphs* (a graph is an interval graph iff there is a 1–1 correspondence between its vertices and a set of intervals on the real line such that two vertices are adjacent iff the corresponding intervals have a nonempty intersection). The *maximal clique vs. vertex matrix* of a graph G has a column for each maximal clique (maximal with respect to set inclusion) and a row for each vertex, with an entry being one iff the vertex is a member of the clique and zero otherwise. It can be shown that a graph is an interval graph iff its maximal clique vs. vertex matrix has the consecutive ones property for rows (see [21]).

2.4. Graph homomorphisms

For directed graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$, G is called H -colorable if there exists a mapping $f: V_G \rightarrow V_H$ such that for all edges $(x, y) \in E_G$ we have $(f(x), f(y)) \in E_H$. The problem of deciding whether some graph G is H -colorable was introduced by Nešetřil [36]. In general, this problem is NP-complete.

Gutjahr et al. [26] introduced so-called X -graphs. A graph is an X -graph, iff there exists an enumeration of its vertices v_1, \dots, v_n such that whenever (v_i, v_j) and (v_k, v_l) are edges, then also $(v_{\min(i,k)}, v_{\min(j,l)})$ is an edge of the graph. Translating this into the language of forbidden submatrices we see that a graph is an X -graph if and only if there exists an enumeration of its vertices such that its adjacency matrix avoids the two submatrices

$$X_1 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad X_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

This is equivalent to saying that the adjacency matrix is in $\mathcal{M}^*(\{X_1, X_2\})$ (of course, the permutations for rows and columns must be identical).

It can be proved [26] that for H an X -graph, the H -coloring problem is solvable in polynomial time. The class of X -graphs contains e.g. all *transitive tournaments* and all *semipaths* (directed paths with arbitrarily directed edges); however, we do not know whether it is possible to recognize X -graphs in polynomial time.

Problem 1. Determine the complexity of membership in $\mathcal{M}^*(\{X_1, X_2\})$, where X_1 and X_2 are defined as above.

We observe that if membership in $\mathcal{M}^*(\{X_1, X_2\})$ can be decided in polynomial time then membership in $\mathcal{M}(\{X_1, X_2\})$ can be decided in polynomial time, too. This can be seen as follows. For M an input matrix for membership in $\mathcal{M}(\{X_1, X_2\})$, we generate a new square matrix M' with sidelength $r(M) + c(M)$. In the upper right part of M' we put matrix M , all other entries of M' are set to "0". It is easily checked that M' is in $\mathcal{M}^*(\{X_1, X_2\})$ if and only if M is in $\mathcal{M}(\{X_1, X_2\})$.

2.5. Connected binary matrices

A matrix $M = (m_{ij})$ is said to be ρ -connected, $\rho \in \mathbb{N}_0$, if for each pair of rows i_1 and i_2 the sequence $\langle m_{i_1, j} - m_{i_2, j} \rangle, j = 1, \dots, c(M)$, has at most ρ sign changes. This class of matrices has attracted a lot of research in the Russian literature due to the fact that for 1-connected and for 2-connected transportation cost matrices the well-known NP-hard *simple plant location problem* becomes polynomial time solvable (see [7, 3]).

With the exception of testing (0, 1)-matrices for being 1-connected, the problem of deciding for a given matrix M and a number $\rho \in \mathbb{N}$ whether there exists a permutation of the columns of M such that the permuted matrix becomes ρ -connected, is still unsolved (cf. [3]).

We note that for $(0, 1)$ -matrices this recognition problem fits well into the framework of our paper. More specifically, a $(0, 1)$ -matrix M is ρ -connected, $\rho \in \mathbb{N}_0$, if and only if it avoids the two $2 \times (\rho + 2)$ matrices $R_1^{(\rho)}$ and $R_2^{(\rho)}$ whose columns are alternatingly $(0, 1)^T$ and $(1, 0)^T$, i.e.

$$R_1^{(\rho)} = \begin{pmatrix} 1 & 0 & 1 & \dots \\ 0 & 1 & 0 & \dots \end{pmatrix} \quad \text{and} \quad R_2^{(\rho)} = \begin{pmatrix} 0 & 1 & 0 & \dots \\ 1 & 0 & 1 & \dots \end{pmatrix}.$$

Beresnev and Davydov [8] settled the case of $\rho = 1$ by giving a polynomial $O(r(M)^2 \cdot c(M)^2)$ time algorithm for solving the membership problem in $\mathcal{N}(\{R_1^{(1)}, R_2^{(1)}\}) = \mathcal{N}(\{R_1^{(1)}, R_2^{(1)}\})$. The case $\rho = 2$ leads to the following open problem.

Problem 2. Determine the complexity of testing a $(0, 1)$ -matrix for membership in the set

$$\mathcal{N}\left(\left\{\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}\right\}\right).$$

2.6. Monge sequences in binary matrices

Let M be an arbitrary matrix and denote by $l = r(M)c(M)$ the number of entries of M . An ordering $\sigma = ((i_1, j_1), (i_2, j_2), \dots, (i_l, j_l))$ of the indices of M is called *Monge sequence* for the matrix M if the following condition is fulfilled:

For every $1 \leq i, p \leq r(M)$, $1 \leq j, q \leq c(M)$, whenever (i, j) precedes both (i, q) and (p, j) in σ , the corresponding entries in matrix M fulfill

$$m_{ij} + m_{pq} \leq m_{iq} + m_{pj}.$$

It is well known that, given an arbitrary ordering σ of the indices (i, j) of a matrix M , a greedy approach which consists in maximizing each variable in turn, yields a *feasible* solution to the Hitchcock transportation problem with M as cost matrix. It has been shown by Hoffman [28] that this solution is also *minimum* for any feasible supply and demand vectors if and only if σ is a Monge sequence for M .

For $(0, 1)$ -matrices M it turns out that a necessary and sufficient condition for the existence of a Monge sequence is that \bar{M} , the complement of M which is obtained by exchanging zeros and ones, has a Γ -free ordering, i.e. $\bar{M} \in \mathcal{N}(\{\Gamma\})$ (see e.g. [29]). This result yields an improvement for constructing a Monge sequence for $(0, 1)$ -matrices over the general algorithm of Alon et al. [4] which runs in $O(r(M)^2 c(M) \log c(M))$ time for an $r(M) \times c(M)$ matrix M . (Here $r(M) \leq c(M)$ is assumed.)

For an application of Monge sequences in $(0, 1)$ -matrices we refer to the work of Adler et al. [1] and Adler and Shamir [2], who investigated so-called *feasibility sequences* in connection with transportation problems on graphs with *forbidden arcs*. (For such problems the greedy method does not necessarily yield a feasible solution.)

2.7. Binary Monge matrices

An $r(M) \times c(M)$ matrix M is called *Monge matrix* if it satisfies

$$m_{ij} + m_{pq} \leq m_{iq} + m_{pj} \quad \text{for all } 1 \leq i < p \leq r(M), 1 \leq j < q \leq c(M). \quad (2)$$

Note that this definition is more restrictive than the definition of a Monge sequence. There are matrices which are not Monge, but for which there exists a Monge sequence. However, for a Monge matrix there exists always at least one Monge sequence, take e.g. the “lexicographical” sequence σ_{lex} which is obtained by placing the entry (i, j) at the $((i-1)c(M) + j)$ th position of σ_{lex} for all $1 \leq i \leq r(M)$, $1 \leq j \leq c(M)$.

Designate by \mathcal{P} the class of *permuted Monge matrices*, i.e. the class of all matrices for which there exists a permutation π_1 of the rows and a permutation π_2 of the columns such that the permuted matrix becomes a Monge matrix. Deineko and Filonenko [20] developed an $O(c(M)(r(M) + \log c(M)))$ recognition algorithm for permuted Monge matrices. (Here again $r(M) \leq c(M)$ is assumed.) Thus it seems that the problem of deciding whether a matrix M can be permuted so as to become Monge can be solved more efficiently than deciding whether or not there is a Monge sequence for M .

Let us now turn to $(0, 1)$ -matrices. From condition (2) it can easily be obtained that a $(0, 1)$ -matrix is Monge iff it avoids the following five submatrices:

$$P_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad P_3 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad P_4 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

$$P_5 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

It follows that a $(0, 1)$ -matrix belongs to the set \mathcal{P} if and only if $M \in \mathcal{M}(\mathcal{F}_p)$ with $\mathcal{F}_p = \{P_1, P_2, P_3, P_4, P_5\}$. In Section 4.3 we will give a very simple characterization of $(0, 1)$ -matrices in \mathcal{P} .

To conclude this subsection, we consider the so-called *bottleneck Monge* property that results from replacing the “+” sign in condition (2) by “max”.

$$\max\{m_{ij}, m_{pq}\} \leq \max\{m_{iq}, m_{pj}\}$$

$$\text{for all } 1 \leq i < p \leq r(M), 1 \leq j < q \leq c(M). \quad (3)$$

A matrix M which fulfills (3) is called *bottleneck Monge matrix*. If the cost matrix of a transportation problem with bottleneck objective function is a bottleneck Monge matrix, the problem can be solved by a greedy algorithm in the same style as in the classical sum case (see e.g. [12, 13]). This result motivates the definition of the class \mathcal{B} of *permuted bottleneck Monge matrices* which contains all matrices which can be permuted to become bottleneck Monge.

Obviously, a $(0, 1)$ -matrix M is bottleneck Monge iff it avoids the following three submatrices:

$$B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad B_3 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Further details on the set $\mathcal{M}(\{B_1, B_2, B_3\})$ will be given in Sections 2.8 and 4.3.

2.8. Bipartite permutation graphs

A directed graph $G = (V, E)$ is called *permutation graph*, if there exists a pair (ρ_1, ρ_2) of permutations of the vertex set V such that there is an edge $(i, j) \in E$ if and only if vertex i precedes vertex j in one of $\{\rho_1, \rho_2\}$ and j precedes i in the other. A *bipartite permutation graph* is a permutation graph which is bipartite.

The following characterization of bipartite permutation graphs follows directly from results of Spinrad et al. [42] and Chen and Yesha [14].

Theorem 2.3. *A bipartite graph $G = (V_1, V_2; E)$ is a permutation graph if and only if its associated $(0, 1)$ -matrix $M(G)$, which has an entry $m_{ij} = 1$ iff $(i, j) \in E$, is a member of the class $\mathcal{M}(\{A_1, A_2, A_3\})$, where*

$$A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Bipartite permutation graphs can be recognized in linear time by applying the algorithm of Spinrad et al. [42]. Hence it follows from Theorem 2.3 that $(0, 1)$ -matrices $M \in \mathcal{M}(\{A_1, A_2, A_3\})$ can be recognized in $O(r(M)c(M))$ time.

Note that combining this result with the observation that the matrices B_1 through B_3 are obtained from the matrices A_1 through A_3 by exchanging the role of zeros and ones, results in an $O(r(M)c(M))$ time recognition algorithm for the class $\mathcal{M}(\{B_1, B_2, B_3\})$ introduced in the previous subsection.

2.9. Grid intersection graphs

Let I_1 and I_2 be finite families of *horizontal* and *vertical* intervals in the plane, such that no two horizontal and vertical intervals intersect. The *intersection graph* $G = (V_1 \cup V_2, E)$ associated with these intervals contains a vertex in V_1 , for every horizontal interval in I_1 and a vertex in V_2 , for every vertical interval in I_2 . Further, there exists an edge $(h, v) \in E$ if and only if the corresponding horizontal interval h and the corresponding vertical interval v intersect.

This bipartite graph G is called a *grid intersection graph* and the family $I_1 \cup I_2$ is called a *grid representation* of G . A $(0, 1)$ -matrix M is said to have a *grid representation* if there exists a grid intersection graph $G = (V_1 \cup V_2, E)$ such that there is an edge $(i, j) \in E$ if and only if $m_{ij} = 1$.

Hartman et al. [27] introduced so-called *cross-free* matrices defined as follows. A $(0, 1)$ -matrix is termed *cross-free* if it avoids the following *cross matrices*:

$$\begin{pmatrix} w & 1 & x \\ 1 & 0 & 1 \\ y & 1 & z \end{pmatrix} \text{ with } w, x, y, z \in \{0, 1\}. \quad (4)$$

The class of $(0, 1)$ -matrices for which there are row and column permutations such that the permuted matrix is cross-free plays an essential role in characterizing the class of matrices which have a grid representation.

Theorem 2.4 (Hartman et al. [27]). *A $(0, 1)$ -matrix M has a grid representation if and only if $M \in \mathcal{M}(\mathcal{F}_c)$, where \mathcal{F}_c contains the cross matrices defined above in (4).*

Unfortunately, however, the problem of deciding whether or not a given $(0, 1)$ -matrix (a given bipartite graph) has a grid representation is NP-complete (cf. [32]), and therefore the same complexity result holds for recognizing matrices in $\mathcal{M}(\mathcal{F}_c)$.

2.10. Some examples with highly symmetric sets of forbidden matrices

A typical example for a situation where highly symmetric sets of forbidden matrices occur is the following problem, which has been dealt with extensively in the graph-theoretical literature. Given an input graph $G = (V_G, E_G)$ and some forbidden graph $H = (V_H, E_H)$, one wants to know whether G contains the forbidden H as induced subgraph or not. We let \mathcal{F}_H contain all feasible permutations of the adjacency matrix of H , where feasible means that the permutation is applied to the rows and columns at the same time. Then G does not contain H as induced subgraph if and only if its adjacency matrix is in $\mathcal{M}^*(\mathcal{F}_H)$.

A $(0, 1)$ -matrix is called *balanced* if it does not contain a square submatrix of odd order with two ones per row and column. This very famous class of $(0, 1)$ -matrices, introduced by Berge [9], plays a fundamental role in connection with the integrality of certain packing and covering polyhedra (see e.g. [9, 35] for a survey). It is easy to see that a totally balanced matrix is also balanced. While as we have seen above, totally balanced matrices can also be described by the set $\mathcal{F} = \{\Gamma\}$ consisting of a single forbidden submatrix, for balanced matrices only a characterization in terms of the above rather inconvenient set of forbidden matrices is known. Until very recently when Conforti et al. [15] gave a polynomial time recognition algorithm for balanced matrices based on deep graph-theoretical insights, it was not even known whether recognizing balancedness belongs to NP.

A closely related subject is that of strongly unimodular matrices. Let \mathcal{F}_s contain all matrices that are forbidden for balanced matrices and additionally all matrices that can be obtained from those matrices by replacing exactly one "0"-entry by a "1"-entry (cf. also [10] for a hypergraph interpretation). A $(0, 1)$ -matrix M is called *strongly*

unimodular (or strongly balanced), see [17], if $M \in \mathcal{M}(\mathcal{F}_\theta)$ (i.e. $M \in \text{AV}(\mathcal{F}_\theta)$). For a simple efficient algorithm for recognizing strongly unimodular matrices the reader is referred to Conforti and Rao [16].

3. Equivalence between different classes of forbidden matrices

Putting together the results reviewed in Sections 2.1 and 2.2 on totally balanced and greedy matrices, we obtain characterizations of the class of totally balanced matrices in terms of three different sets of forbidden submatrices. Let $\mathcal{F}_{\mathcal{F}_1} = \bigcup_{k \geq 3} \mathcal{A}_k$ be the class of cycle submatrices defined above, and let $\mathcal{F}_{\mathcal{F}_2} = \{F\}$ and $\mathcal{F}_{\mathcal{F}_3} = \{G_1, G_2\}$, where G_1 and G_2 denote again the two forbidden matrices defined in Section 2.2 on greedy matrices. Then we have the following equivalence result (cf. Theorems 2.1 and 2.2):

$$\mathcal{M}(\mathcal{F}_{\mathcal{F}_1}) = \mathcal{M}(\mathcal{F}_{\mathcal{F}_2}) = \mathcal{M}(\mathcal{F}_{\mathcal{F}_3}).$$

All these sets describe the class of totally balanced matrices. Hence, from a general point of view, the following question arises.

Problem 3. For given classes of forbidden matrices \mathcal{F}_1 and \mathcal{F}_2 , is it Turing-decidable whether $\mathcal{M}(\mathcal{F}_1) = \mathcal{M}(\mathcal{F}_2)$ holds?

For the sake of convenience let us call the forbidden sets \mathcal{F}_1 and \mathcal{F}_2 *equivalent* if they represent the same class of matrices, i.e. if $\mathcal{M}(\mathcal{F}_1) = \mathcal{M}(\mathcal{F}_2)$. In the sequel we will exhibit a special class of equivalent sets of forbidden submatrices. For that purpose we generalize the arguments used by Hoffman, Kolen and Sakarovitch to prove Theorem 2.2 and obtain the following general theorem.

Theorem 3.1. Let F be a $2 \times k$ $(0,1)$ -matrix with the property that its last column is either equal to $(1,0)^T$ (case 1) or to $(0,1)^T$ (case 2). Let F_1^* denote the matrix that is obtained from F by appending a $(k+1)$ st column equal to $(0,1)^T$ in case 1 and equal to $(1,0)^T$ in case 2, and let F_2^* be the matrix that results from F_1^* by exchanging the two rows of F_1^* .

If F is a submatrix not only of F_1^* but also of F_2^* then we have

$$\mathcal{M}(\{F\}) = \mathcal{M}(\{F_1^*, F_2^*\}) = \mathcal{M}^c(\{F_1^*, F_2^*\}). \quad (5)$$

Proof. Since F is a submatrix of F_1^* and F_2^* by assumption, it is trivial that $\mathcal{M}(\{F\}) \subseteq \mathcal{M}(\{F_1^*, F_2^*\})$. To prove the other direction, we will show that, given a $(0,1)$ -matrix M which avoids both F_1^* and F_2^* , the rows of M can be permuted such that the resulting matrix avoids F . Assume that the last column of F is $(1,0)^T$. (The other case can be treated analogously.) Let $M \in \text{AV}(\{F_1^*, F_2^*\})$. We now regard the rows of M as vectors (read from right to left) and sort them lexicographically

increasing. Call the resulting matrix \tilde{M} . Thanks to the invariance of the set $\{F_1^*, F_2^*\}$ to row permutations, this matrix still avoids both F_1^* and F_2^* .

Now suppose that F occurs as submatrix in \tilde{M} within rows i_1, i_2 ($i_1 < i_2$) and columns j_1, j_2, \dots, j_k ($j_1 < j_2 < \dots < j_k$). Since the rows are ordered lexicographically, we know that there exists a column j_{k+1} ($j_{k+1} > j_k$) such that $\tilde{m}_{i_1, j_{k+1}} = 0$ and $\tilde{m}_{i_2, j_{k+1}} = 1$. But this contradicts the fact that \tilde{M} avoids F_1^* . Thus we get the desired relation $\mathcal{M}(\{F_1^*, F_2^*\}) \subseteq \mathcal{M}(\{F\})$. \square

From the theorem above not only the identity (1) already proved in [30] (set $F := \{F\}$) can be deduced, among others one immediately obtains the following equivalence result.

Corollary 3.2. Let $R_1^{(\rho)}$ and $R_2^{(\rho)}$ denote the “Russian” matrices defined in Section 2.5 on connected matrices. Then for all $\rho \in \mathbb{N}$ we have

$$\mathcal{M}(R_1^{(\rho-1)}) = \mathcal{M}(\{R_1^{(\rho)}, R_2^{(\rho)}\}) = \mathcal{M}^c(\{R_1^{(\rho)}, R_2^{(\rho)}\}). \quad (6)$$

Proof. Set $F = R_1^{(\rho-1)}$ and apply Theorem 3.1. \square

Observe that in view of the above result Problem 2 of recognizing (0, 1) 2-connected matrices can be rephrased as testing a (0, 1)-matrix for membership in the set

$$\mathcal{M}\left(\left\{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}\right\}\right).$$

Since the equivalence results presented above are of a very special structure, further results in this direction would be of great importance.

Problem 4. Find further examples for classes of equivalent sets \mathcal{F}_1 and \mathcal{F}_2 , in particular for matrices with more than two different entries.

4. Small matrices

In this section we determine the complexity of membership in $\mathcal{M}(\mathcal{F})$ when \mathcal{F} contains a single 1×3 or 2×2 matrix. Furthermore we present some partial results on sets \mathcal{F} containing more than just one 2×2 matrix.

4.1. One-by-three matrices

For reasons of symmetry, we only have to investigate the four matrices depicted in the left column of Table 1.

Avoiding (0, 0, 0): Dealing with (0, 0, 0) is trivial. In case (0, 0, 0) appears as submatrix of some matrix M , it cannot be avoided by permuting columns or rows. Hence, membership in $\mathcal{M}(\{(0, 0, 0)\})$ reduces to counting the number of zeros in every row of M .

Table 1
Complexity results for 1×3 matrices

Matrix F	Membership in $\mathcal{H}(\{\mathcal{F}\})$	Argument
(0,0,0)	Polynomial for arbitrary matrices	Trivial
(0,1,1)	Polynomial for arbitrary matrices	Lemma 4.1
(1,0,1)	Polynomial for (0,1)-matrices	PQ -trees
	NP-complete for (0,2)-matrices	Theorem 5.1
(0,1,2)	Open for (0,2)-matrices	???
	NP-complete for (0,3)-matrices	Theorem 5.1

Avoiding (1,0,1): In Section 2.3 on the consecutive ones property we already mentioned that membership of (0,1)-matrices in $\mathcal{H}(\{(1,0,1)\})$ can be checked in linear time by applying PQ -trees, a data structure developed by Booth and Lueker [11].

A PQ -tree is able to store special sets of *permissible* permutations of a set S while using only linear storage. The permissible permutations are those in which certain subsets $S' \subset S$ occur as consecutive subsequences. As the elements of a new subset S' are constrained to appear together, the number of permissible permutations is reduced. Booth and Lueker show how to perform such a reduction in an efficient, polynomial way. Obviously, this yields a polynomial time recognition algorithm for (0,1)-matrices with the consecutive ones property.

In Theorem 5.1 it will be proved, however, that recognizing matrices in $\mathcal{H}(\{(1,0,1)\})$ becomes NP-complete for (0,2)-matrices.

Avoiding (0,1,1): For this 1×3 matrix there exists a simple polynomial time algorithm which works for arbitrary matrices M .

Lemma 4.1. *There exists a polynomial time algorithm for checking whether some (arbitrary) matrix M is in $\mathcal{H}(\{(0,1,1)\})$.*

Proof. If there exists a row i in M containing no zero or at most one entry 1, this row cannot produce a submatrix (0,1,1); hence, we may remove i from M . If there exists a column j in M not containing any entry 0, we can make it the leftmost column and j cannot contribute to any submatrix (0,1,1). Thus, we may remove j from M without changing the answer to our problem. We call a matrix M *reduced* if it neither contains rows nor columns that may be removed this way.

We claim that a reduced matrix M is in $\mathcal{H}(\{(0,1,1)\})$ if and only if M is the empty matrix. Assume the contrapositive, i.e. that there exists a reduced nonempty matrix M_0 in $\mathcal{AV}(\{(0,1,1)\})$. Consider the leftmost column in M_0 . This column contains at least one entry 0, say in row i_0 . Row i_0 contains at least two entries 1, both lying to the right of the entry 0; a contradiction.

Detecting removable rows and columns is easily done in polynomial time. This immediately yields the claimed polynomial time algorithm for (0,1,1). \square

Avoiding (0, 1, 2): The claimed NP-completeness result for (0, 3)-matrices will be proven in Theorem 5.1. The only open problem concerning membership complexity for 1×3 matrices is the following problem.

Problem 5. Determine the complexity of recognizing the class of (0, 2)-matrices belonging to the set $\mathcal{M}^c(\{(0, 1, 2)\})$.

4.2. Two-by-two matrices

In this section we will only consider (0, 1)-matrices. For reasons of symmetry, there are only four combinatorially different 2×2 (0, 1)-matrices:

$$S_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad S_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad S_4 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}.$$

Avoiding matrix S_1 : Matrices with no S_1 as submatrix are called *linear* by Berge [10]. Matrix S_1 leads to an easy problem, as this matrix is invariant under permutations. Checking all 2×2 submatrices of the input matrix M yields a trivial polynomial time algorithm. However, there exist faster methods to solve this problem.

Theorem 4.2. For M a (0, 1)-matrix with $f(M)$ entries equal to one, it can be checked in $O(r(M)c(M) + \min\{r(M)^2, c(M)^2, f(M)^{3/2}\})$ time whether M contains S_1 .

Proof. The claimed result follows from a combination of two algorithms with running times of $O(r(M)c(M) + f(M)^{3/2})$ resp. $O(r(M)c(M) + \min\{r(M)^2, c(M)^2\})$, whichever is faster.

The $O(r(M)c(M) + f(M)^{3/2})$ time algorithm is a direct consequence of a graph-theoretical result of de Berg and van Kreveld [18] where it is shown that in a bipartite graph with q edges a cycle of length four can be found in $q^{3/2}$ time. (Several related results may be found in [19].) Obviously, checking whether M avoids S_1 reduces to testing whether the bipartite graph $B(M)$ which is associated with M contains a cycle of length four. Since $B(M)$ has $f(M)$ edges and it takes $O(r(M)c(M))$ time to construct $B(M)$ from the given matrix M , the claimed time bound follows immediately.

The $O(r(M)c(M) + \min\{r(M)^2, c(M)^2\})$ time algorithm is based on the following simple idea due to Spinrad [41]: W.l.o.g. we assume $r(M) \leq c(M)$ (otherwise we take M^T). Let $A(M) = (a_{ip})$ be an $r(M) \times r(M)$ matrix such that a_{ip} , $i \neq p$, equals the number of columns which have a 1 entry both in row i and row p . Obviously, M avoids S_1 iff $a_{ij} < 2$ for all $1 \leq i < p \leq n$. $A(M)$ is constructed incrementally by the following algorithm: First we initialize $A(M)$ to the zero matrix. Then we scan the columns one after another. For any two rows i and p , $i < p$ which have a 1 in the current column j we add 1 to the entry a_{ip} . As soon as an entry of $A(M)$ becomes 2, we stop. It takes at most $O(r(M)c(M) + r(M)^2)$ steps until either the whole matrix M has been scanned or one entry of $A(M)$ has become 2. \square

Avoiding matrix S_4 : The case of S_4 is again easy to solve.

Lemma 4.3. *It can be checked in $O(r(M)c(M)^2)$ time whether a given $r(M) \times c(M)$ $(0, 1)$ -matrix M is in $\mathcal{H}(\{S_4\})$.*

Proof. In order to avoid S_4 , row permutations are useless. If S_4 appears within two columns j_1 and j_2 , j_2 must precede j_1 in any legal ordering of the matrix. This yields a partial order on the columns of M . M is in $\mathcal{H}(\{S_4\})$ if and only if this partial order can be embedded into a total order.

This algorithm can be implemented in $O(r(M)c(M)^2)$ time. (We need $O(r(M)c(M)^2)$ time to construct the partial order since there are $\binom{c(M)}{2}$ pairs of columns to be considered. The final check whether the partial order can be embedded into a total order takes $O(c(M)^2)$ time.) \square

Avoiding matrix S_2 : Note that $S_2 = \Gamma$. In Section 2 we have seen that there are numerous applications for which it is important to be able to recognize matrices in $\mathcal{H}(\{\Gamma\})$. In the sequel we will describe the main ideas which are needed for recognizing this class of matrices.

A matrix M is called *totally reverse lexicographic* (TRL) or *doubly lexical* [35, 34] if the rows and columns of M are ordered such that both the row vectors (read from right to left) and the column vectors (read from bottom to top) are *increasing* with respect to the usual lexicographic (“dictionary”) order.

Theorem 4.4 (Lubiw [34], Hoffman et al. [30]). *Let M be a $(0, 1)$ -matrix and \bar{M} be an arbitrary totally reverse lexicographic ordering of M . Then the following conditions are equivalent:*

- (i) $\bar{M} \in \text{AV}(\{\Gamma\})$ (i.e. \bar{M} is Γ -free),
- (ii) $M \in \mathcal{H}(\{\Gamma\})$ (i.e. M has an Γ -free ordering).

Thus we need an efficient algorithm for determining a TRL ordering of a $(0, 1)$ -matrix M . The algorithm of Paige and Tarjan [38] which relies on ideas of Lubiw [34] accomplishes this task in $O(L \log L)$ time and $O(L)$ space with $L = r(M) + c(M) + f(M)$ provided that M is given as list of its entries which are equal to one. ($f(M)$ again denotes the number of ones in M .) Recently Spinrad [40] obtained an alternative double lexical ordering algorithm which runs in $O(r(M)c(M))$ time. Spinrad’s algorithm is hence well suited for dense matrices with a large number of ones.

Furthermore, Lubiw [34] gave a linear time algorithm for testing the Γ -freeness of a $(0, 1)$ -matrix M in TRL order which runs in $O(L)$ time. (The straightforward approach of checking all 2×2 submatrices of M obviously does not have this nice property.)

Avoiding matrix S_3 : At first sight, Matrix S_3 , the only 2×2 matrix not discussed so far, seems to be rather hard to tackle. However, it is easy to see that a matrix that has

the consecutive ones property for rows (cf. Section 2.3) can be permuted to avoid S_3 , e.g. by sorting the rows such that the positions of the first nonzero entry in each row form a nonincreasing sequence.

An analogous procedure succeeds if M has the consecutive ones property for columns, or the consecutive zeros property for rows or columns, respectively. The following example matrix M shows, however, that having one of these four consecutive properties is not a necessary condition for a matrix to belong to $\mathcal{M}(\{S_3\})$:

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Due to the 3×3 submatrices in the upper left corner and in the lower right corner of M , M has none of the four consecutive properties, but M avoids S_1 as can easily be checked by inspection.

Nevertheless, the membership problem in $\mathcal{M}(\{S_3\})$ can be solved in polynomial time. This result relies on the equivalence result in Corollary 3.2. Note that for $\rho = 1$ relation (6) yields that the sets $\mathcal{F}_1 = \{S_3\}$ and $\mathcal{F}_2 = \{R_1^{(1)}, R_2^{(1)}\}$ are equivalent. Hence, we may use the algorithm of Beresnev and Davydov [8] mentioned in Section 2.5 which solves the problem for \mathcal{F}_2 to solve our problem for \mathcal{F}_1 . It follows that the membership problem in $\mathcal{M}(\{S_3\})$ can be solved in $O(r(M)^2 c(M)^2)$ time.

We mention that a faster algorithm for membership in $\mathcal{M}(\mathcal{F}_1)$ would also yield an improved algorithm for recognizing 1-connected matrices (cf. Section 2.5) and for solving the simple plant location problem for this class of matrices (cf. [3]).

4.3. Sets of small forbidden matrices based on Monge matrices

The investigations in this section are motivated by the sets of forbidden submatrices that occur in recognizing $(0, 1)$ -matrices that belong to the classes of permuted Monge matrices and permuted bottleneck Monge matrices, respectively.

(0, 1) Monge matrices: Recall that for $(0, 1)$ Monge matrices the set \mathcal{F}_ρ of forbidden submatrices consists of the following five forbidden 2×2 matrices:

$$P_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad P_3 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad P_4 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

$$P_5 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

As already mentioned in Section 2.7, the algorithm of Deineko and Filonenko [20] solves the membership problem in $\mathcal{M}(\mathcal{F}_p)$. For $(0, 1)$ -matrices M this algorithm runs in $O(r(M)c(M))$ time. The theorem below can be used to derive an alternative polynomial time algorithm, but its main importance is to show that matrices in $\mathcal{M}(\mathcal{F}_p)$ are of a very special structure. More specifically, after deleting multiply occurring rows and columns, the largest order of a matrix $M \in \mathcal{M}(\mathcal{F}_p)$ is 4×3 or 3×4 , respectively.

Theorem 4.5. *Let M be a $(0, 1)$ -matrix which contains no identical rows and columns and satisfies $r(M) \leq c(M)$. Then the following two properties hold:*

- (i) *If $M \in \mathcal{M}(\mathcal{F}_p)$, then M has at most 3 rows and at most 4 columns.*
- (ii) *Let M be a 3×4 matrix within the class $\text{AV}(\mathcal{F}_p)$. Then M must be equal to one of the following four matrices:*

$$Q_1^a = \begin{pmatrix} 0 & 0 & 1 & 1 \\ a & 0 & 1 & a \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad (a = 0, 1), \quad Q_2^b = \begin{pmatrix} 0 & 1 & 0 & 1 \\ b & 1 & 0 & b \\ 1 & 1 & 0 & 0 \end{pmatrix}, \quad (b = 0, 1).$$

Proof. (i) Let $Y = \{y^{(1)}, \dots, y^{(d)}\}$, $d = 2^{r(M)}$, be the set of all distinct $(0, 1)$ vectors which might appear as column vectors in a matrix with $r(M)$ rows. Let $G_c = (V_c, E_c)$ define a directed graph with vertex set $V_c = Y$, where two vertices $y, y' \in V_c$, $y \neq y'$, are joined by an edge $(y, y') \in E_c$ if and only if

$$y_k + y_{k+1} \leq y_{k+1} + y'_k \quad \text{for all } k = 1, \dots, r(M) - 1. \quad (7)$$

If property (7) holds, the column vector y may be placed to the left of the column vector y' without violating the Monge property (2).

A vertex $y \in V_c$ is said to be a *source* resp. a *sink* of G_c if there is no edge in E_c which enters resp. leaves y . All remaining vertices are called *intermediary vertices*. The following two observations are crucial to the proof that $c(M) \leq 4$.

- (a) A vertex $y \in Y$ is a source if and only if $y_1 = 0$ and $y_{r(M)} = 1$, a sink if and only if $y_1 = 1$ and $y_{r(M)} = 0$ and an intermediary vertex if and only if $y_1 = y_{r(M)}$.
- (b) If y and y' are two intermediary vertices such that $(y, y') \in E_c$, then one of y and y' must be the zero vector all of whose components are zero and the other must be the all ones vector all of whose components are one.

To prove (a), note that the only column which can be placed to be the left of a column y with $y_1 = 0$ and $y_{r(M)} = 1$ without violating property (7) is y itself. Hence, y is a source. Conversely, if y is column such that $y_1 \neq 0$ or $y_{r(M)} \neq 1$, then we can easily find a column y' such that $(y', y) \in E_c$ (if $y_1 \neq 0$ we set $y'_1 = 0$ and $y'_i = y_i$, $i = 2, \dots, r(M)$, the case $y_{r(M)} \neq 1$ is dealt with analogously). Hence y cannot be a source. The proof for a sink is symmetric. Property (b) can be shown as follows: Since y and y' are intermediary vertices we have $y_1 = y_{r(M)}$ and $y'_1 = y'_{r(M)}$. We now distinguish the following two cases:

Case 1: $y_i = y'_i$. Since y and y' need to be pairwise distinct there exists an index i such that $y_i \neq y'_i$, i.e. either we have $y_i = 1$ and $y'_i = 0$ or $y_i = 0$ and $y'_i = 1$. It is easy to see that both subcases lead to a contradiction to condition (7).

Case 2: $y_i \neq y'_i$. It can easily be checked that there cannot exist an index i such that $y_i \neq y_1$ or $y'_i \neq y'_1$. In both subcases we again arrive at a contradiction to (7). Consequently, one of y and y' must be the zero vector while the other is the all ones vector.

It immediately follows from (b) that $c(M) \leq 4$. Since $r(M) \leq c(M)$ by assumption this implies $r(M) \leq 4$. Hence in order to prove $r(M) \leq 3$, it remains to be shown that the case $c(M) = 4$ and $r(M) = 4$ cannot arise. This case can only occur if the first column of M is a source, the zero vector and the all ones vector are the two middle columns of M and the last column is a sink. Since all entries of the second and third column, respectively, are identical, two rows i_1 and i_2 which are distinct either fulfill $m_{i_1,1} \neq m_{i_2,1}$ or $m_{i_1,4} \neq m_{i_2,4}$. Since the first column is a source and the last column is a sink, we have $m_{i_1,1} = 0$ and $m_{i_1,4} = 1$ and $m_{i_2,1} = 1$ and $m_{i_2,4} = 0$. It is now easy to see by inspection that the remaining two cases that M contains a row i such that $m_{i,1} = m_{i,4} = 0$ and a row p such that $m_{p,1} = m_{p,4} = 1$ cannot occur at the same time (one arrives at a contradiction to $M \in \text{AV}(\mathcal{F}_g)$). Thus $r(M) \leq 3$.

(ii) Follows immediately from the considerations in the last paragraph. The matrices Q_1^0 and Q_1^1 are obtained if the zero vector precedes the all ones vector, while the matrices Q_2^0 and Q_2^1 result when the all ones vector precedes the zero vector. \square

The characterization of matrices in the set $\mathcal{M}(\mathcal{F}_g)$ given in Theorem 4.5 above demonstrates on the basis of $(0,1)$ -matrices that in general the class of matrices which can be permuted to become Monge matrices is considerably smaller than the class of matrices for which there exists a Monge sequence (cf. also Sections 2.6 and 2.7).

(0, 1) *bottleneck Monge matrices:* Recall that for $(0, 1)$ bottleneck Monge matrices the set \mathcal{F}_g of forbidden submatrices contains the following three 2×2 matrices:

$$B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad B_3 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

In the sequel we present some results on the set $\mathcal{M}(\mathcal{F}_g)$, i.e. the class of permuted $(0, 1)$ bottleneck Monge matrices.

We start with two characterizations of the set $\mathcal{M}(\mathcal{F}_g)$. The first one, a graph-theoretical characterization, is a direct consequence of Theorem 2.3 and states that a $(0, 1)$ -matrix M belongs to $\mathcal{M}(\mathcal{F}_g)$ if and only if its associated bipartite graph $B(M)$ is the complement of a bipartite permutation graph. Consequently, a $(0, 1)$ -matrix $M \in \mathcal{M}(\mathcal{F}_g)$ can be recognized in $O(r(M)c(M))$ time by applying the algorithm of Spinrad et al. [42] which recognizes bipartite permutation graphs in linear time (cf. Section 2.8).

The second characterization describes the set $\mathcal{M}(\mathcal{F}_g)$ directly in matrix terms. Let s_i resp. f_i denote the position of the first resp. last zero in row i . A $(0, 1)$ -matrix M is said to be a *double staircase matrix* if $s_1 \leq s_2 \leq \dots \leq s_{r(M)}$, $f_1 \leq f_2 \leq \dots \leq f_{r(M)}$ and

$m_{ij} = 0$ for all $j \in [s_i, f_i]$, i.e. if the zeros in each row are consecutive and if the rows are ordered increasingly with respect to both the first and the last zero entry in each row. The term “double staircase” is introduced since in a pictorial setting the positions of the first resp. last zero in each row form a staircase.

The following theorem relates double staircase matrices and (0,1) bottleneck Monge matrices.

Theorem 4.6 (Klinz et al. [31]). *A (0,1)-matrix M with no rows and columns of all ones belongs to the set $\mathcal{M}(\mathcal{F}_d)$ if and only if there exists a permutation π_1 of its rows and a permutation π_2 of its columns such that the permuted matrix is a double staircase matrix.*

In [31] an algorithm is presented which, given a (0,1)-matrix M , determines in $O(r(M)c(M))$ time a pair (π_1, π_2) of row and column permutations such that the permuted matrix M_{π_1, π_2} is a double staircase matrix or proves that no such pair exists. In connection with Theorem 4.5 above this gives another $O(r(M)c(M))$ time algorithm for recognizing matrices $M \in \mathcal{M}(\mathcal{F}_d)$. In contrast to the graph-theoretical algorithm of Spinrad et al. [42] this algorithm directly addresses the matrix problem and delivers the set of all pairs (π_1, π_2) of row and column permutations such that the permuted matrix M_{π_1, π_2} is a bottleneck Monge matrix.

Note that in every double staircase matrix the zeros in each row and in each column are consecutive. Thus it follows from Theorem 4.6 that the consecutive zeros property for rows and for columns is a necessary condition for a (0,1)-matrix to belong to $\mathcal{M}(\mathcal{F}_d)$. The example matrix

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

which is not in $\mathcal{M}(\mathcal{F}_d)$ shows, however, that this condition is not sufficient. Tucker [43] has given a forbidden submatrix characterization of all (0,1)-matrices with the consecutive zeros property for both rows and columns. This leads to asking which matrices have to be added to Tucker's set of forbidden submatrices in order to get a characterization of the set $\mathcal{M}(\mathcal{F}_d)$. This question has recently been answered in [31] where the following theorem is shown.

Theorem 4.7. *Let M be a (0,1)-matrix which has the consecutive zeros property for both rows and columns. Let \mathcal{M} denote the set of all matrices which can be obtained from the matrices B_4, B_4^T, B_5 and B_5^T with*

$$B_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}, \quad B_5 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

by permuting rows and columns.

Then M is a permuted 0-1 bottleneck Monge matrix if and only if M does not contain any submatrix from the set \mathcal{A} .

As immediate corollary of Theorem 4.7 and Tucker's forbidden submatrix characterization we obtain the following equivalence result.

Corollary 4.8. *The sets $\{B_1, B_2, B_3\}$ and $\{B_2, B_3\}$ are equivalent, i.e. in other words,*

$$\mathcal{M}(\{B_1, B_2, B_3\}) = \mathcal{M}(\{B_2, B_3\}).$$

Proof. It is straightforward to check by inspection that B_4, B_5 and all forbidden submatrices of Tucker cannot be permuted to avoid B_2 and B_3 at the same time. Hence they do not belong to the set $\mathcal{M}(\{B_2, B_3\})$. \square

Interestingly, the set $\mathcal{M}(\mathcal{F}_a)$ of permuted (0,1) bottleneck Monge matrices plays also a role in the recognition of permuted bottleneck Monge matrices with arbitrary entries (not restricted to 0 and 1) as is shown by the following observation from [31].

Observation 4.9. *Let $\tilde{m}_1 > \tilde{m}_2 > \dots > \tilde{m}_l$ be the pairwise distinct values of entries of matrix M and associate with each value \tilde{m}_k , $1 \leq k \leq l$, a (0,1) matrix T^k which is constructed as follows: If entry $m_{ij} \leq \tilde{m}_k$ then the corresponding entry in T^k is 0, otherwise it is 1.*

Then M is a bottleneck Monge matrix if and only if all matrices T^k , $k = 1, \dots, l$, are bottleneck Monge matrices.

Klinz et al. [31] used Observation 4.9 to derive a polynomial time algorithm which decides in $O(r(M)^2 c(M) + r(M)c(M)^2)$ time whether a given matrix M with arbitrary entries is a permuted bottleneck Monge matrix. The basic idea of this algorithm is to compute for each (0,1)-matrix T^k , $k = 1, \dots, l$, the set of all pairs of row and column permutations which transform T^k into a bottleneck Monge matrix. In a final step it is checked whether there exists a common pair (π_1, π_2) of row and column permutations which belongs to all k sets.

5. NP-completeness results

In this section we will present three NP-completeness results. The NP-completeness proofs for the recognition of (0,2)-matrices with forbidden submatrix (1,0,1) and for (0,3)-matrices with forbidden submatrix (0,1,2) both are straightforward and use similar reductions. As a main result we present a set \mathcal{F} of two (0,1)-matrices, such that deciding membership of a (0,1)-matrix in $\mathcal{M}(\mathcal{F})$ is NP-complete.

Theorem 5.1. For $\mathcal{F}_1 = \{(1, 0, 1)\}$ and $\mathcal{F}_2 = \{(0, 1, 2)\}$, it is NP-complete to decide whether

- (i) a (0,2)-matrix belongs to $\mathcal{M}(\mathcal{F}_1)$, and whether
- (ii) a (0,3)-matrix belongs to $\mathcal{M}(\mathcal{F}_2)$.

Proof. We use reductions to the NP-complete BETWEENNESS problem (cf. [37] resp. [23], Problem [MS1]) that is defined as follows.

INSTANCE: A finite set A and a collection T of ordered triples (a, b, c) of distinct elements from A .

QUESTION: Does there exist an ordering of the elements in A such that for each $(a, b, c) \in T$, element b lies between a and c ?

For each instance of BETWEENNESS, we will construct matrices M_1 and M_2 such that M_i is in $\mathcal{M}(\mathcal{F}_i)$ if and only if the BETWEENNESS instance is satisfiable.

Our matrix M_1 consists of $|A|$ columns corresponding to the elements of A . For each triple (a, b, c) in T , we introduce two rows in M_1 as follows. Both rows consist of $|A| - 3$ entries 2, of two entries 1 and of one entry 0. The first row has its entry 0 at the crossing with column a and the two entries 1 at the crossings with columns b and c . (This row forbids all permutations with a between b and c .) The second row has its entry 0 at the crossing with column c and the two entries 1 at the crossings with columns a and b . (This row forbids all permutations with c between a and b .)

In case M_1 is in $\mathcal{M}(\mathcal{F}_1)$, the rows introduced for each triple (a, b, c) force that column b is placed between columns a and c . Vice versa, a legal ordering of A yields a permutation of M_1 in $\mathcal{AV}(\mathcal{F}_1)$. This settles the proof of (i).

Matrix M_2 consists of $|A|$ columns (corresponding to the elements in A) and of $4|T|$ rows. For each triple $(a, b, c) \in T$ we introduce four rows with entries 0-2-1, 1-2-0, 1-0-2 and 2-0-1 at the crossings with columns a, b and c and with all other entries equal to 3. (Note that in this case we need four rows instead of two to exclude all cases where a is between b and c or c is between a and b . The reason is that \mathcal{F}_2 contains three distinct entries, while \mathcal{F}_1 contains only two distinct entries.) Analogously to the proof of (i), we see that a legal ordering of A exists if and only if M_2 is in $\mathcal{M}(\mathcal{F}_2)$. \square

$$A = (1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1), \quad B = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$$

Fig. 1. Two (0,1)-matrices that gives rise to an NP-complete permutation problem.

Next, we deal with forbidden sets of $(0, 1)$ -matrices. We define $(0, 1)$ -matrices A and B as shown in Fig. 1.

Theorem 5.2. For $\mathcal{F} = \{A, B\}$, it is NP-complete to decide whether a $(0, 1)$ -matrix belongs to $\mathcal{M}(\mathcal{F})$.

Proof. We will use a transformation from the well-known NP-complete 3-SATISFIABILITY problem (cf. [23]), 3SAT for short. This problem is defined as follows.

INSTANCE: A set U of variables; a collection C of clauses over U , such that each $c \in C$ consists of exactly three literals and such that each variable appears (negated and unnegated) at most five times in the clauses in C .

QUESTION: Does there exist a satisfying truth assignment for C (i.e. such that each clause in C contains at least one true literal)?

We fix some instance (U, C) of 3SAT and we construct a corresponding matrix M with $12|U| + |C|$ rows and $3|U|$ columns. Let $\{x_1, x_2, \dots, x_n\}$ be an enumeration of the variables in U .

For each variable $x_i \in U$, we introduce three columns $c_1(i)$, $c_2(i)$ and $c_3(i)$ and we specify the first $12|U|$ entries in these columns as follows. The first $12i - 6$ entries in $c_1(i)$ and the first $12i$ entries in $c_2(i)$ and $c_3(i)$ are set to "1", all other entries in the first $12|U|$ rows are set to "0". These upper $12|U|$ rows of the matrix are called the *control part* of M . This control part ensures that in any legal ordering of M that avoids B , the three columns corresponding to variable x_i must precede the three columns corresponding to variable x_{i+1} and the column $c_1(i)$ must precede columns $c_2(i)$ and $c_3(i)$. In other words, the leftmost column must be $c_1(1)$, followed by $c_2(1)$ and $c_3(1)$ in *arbitrary order*, followed by $c_1(2)$, then $c_2(2)$ and $c_3(2)$ in arbitrary order, and so on.

For each clause we generate a row corresponding to this clause. This row consists of six entries "1" and $3|U| - 6$ entries "0": For each of the three variables x_i in the clause, we create an entry "1" at the crossing of the clause row with the corresponding column $c_1(i)$. In case x_i appears unnegated in the clause, we put a "1" at the crossing with $c_2(i)$ and a "0" at the crossing with $c_3(i)$. In case x_i appears negated in the clause, we put a "0" at the crossing with $c_2(i)$ and a "1" at the crossing with $c_3(i)$.

We claim that M allows an ordering avoiding submatrices A and B if and only if the 3SAT instance is satisfiable.

First, assume that there exists a legal ordering for M . We construct a truth assignment for the variables in U in the following way: If column $c_2(i)$ is to the left of $c_3(i)$ then x_i receives the value TRUE, otherwise it is set to FALSE. Since M avoids B , the control part enforces that for each variable the three corresponding columns form a contiguous submatrix. Hence, a false literal in a clause produces a contiguous submatrix $(1, 0, 1)$ in the corresponding clause row. If there is a clause in C that is FALSE under our truth assignment, the three false literals would produce three

contiguous submatrices $(1, 0, 1)$ in the same clause row, and the ordering would contain A as submatrix. Therefore, every clause is satisfied.

Now assume that there exists a satisfying truth assignment for the 3SAT instance. We describe how to sort the columns of M (permuting the rows cannot avoid or produce submatrices A or B). For $1 \leq i \leq n$, we let $c_1(i) < c_2(i)$, $c_3(i)$ and for $1 \leq i \leq n - 1$, we let $c_2(i), c_3(i) < c_1(i + 1)$. For x_i TRUE, we set $c_2(i) < c_3(i)$ and for x_i FALSE, we set $c_3(i) < c_2(i)$.

- First suppose that our ordering of M contains submatrix B . Any entry “1” belonging to the control part has only “1”-entries to its right. Thus, B must be solely formed by entries from the clause rows. But each variable appears in at most five clauses, and there are at most five entries “1” in every column outside the control part. Consequently, our ordering avoids B .
- Next suppose that our ordering of M contains submatrix A . The “1”-entries in the control part form contiguous blocks in every row. Hence, one of the clause rows must contain A . But now it is easy to see that this implies the existence of a clause with three false literals, a contradiction to the satisfying truth assignment.

This contradiction completes the proof of the theorem. \square

Since the set \mathcal{F} used in the proof of Theorem 5.2 contains *two* matrices the following natural question arises.

Problem 6. Does there exist a single $(0, 1)$ -matrix F , such that the membership problem for $(0, 1)$ -matrices in $\mathcal{M}(\{F\})$ is NP-complete?

Next we discuss the effects of allowing M to contain entries different from those in \mathcal{F} . In the proof of Theorem 5.1, the number “2” (respectively, “3”) was allowed to appear in M , but it did not appear within the forbidden submatrices. We could abuse it as a *dummy entry* to fill up all the positions not explicitly used in our construction. In the proof of Theorem 5.2, we did not have dummy entries at our disposal, and our arguments became more complicated. We conjecture that in general dummy entries make the membership problem NP-complete (but the matrix $(0, 1, 1)$ discussed in Lemma 4.1 demonstrates that this is not always the case).

Problem 7. Prove or disprove: If F is a $(0, e)$ -matrix ($e \geq 2$) with at least three pairwise distinct entries, then the membership problem for $(0, e + 1)$ -matrices in the set $\mathcal{M}(\{F\})$ is NP-complete.

It seems possible that the recognition problem stated in Problem 7 remains hard even when the set of input matrices is restricted to $(0, e)$ -matrices.

Problem 8. Prove or disprove: If F is a $(0, e)$ -matrix ($e \geq 2$) with at least three pairwise distinct entries, then the membership problem for $(0, e)$ -matrices in the set $\mathcal{M}(\{F\})$ is NP-complete.

Finally, we observe that in order to avoid the matrices A and B in Fig. 1 only column permutations are needed whereas row permutations are useless. Let A^T and B^T denote the transposed matrices of A and B . Then membership in $\mathcal{M}^c(\{A^T, B^T\})$ is polynomial time decidable (column permutations cannot avoid A^T and B^T), whereas membership in $\mathcal{M}(\{A^T, B^T\})$ is NP-complete. This leads to the following (inverse) problem.

Problem 9. Does there exist a set \mathcal{F} of matrices such that membership in $\mathcal{M}^c(\mathcal{F})$ is NP-complete whereas membership in $\mathcal{M}(\mathcal{F})$ can be decided in polynomial time?

6. Discussion

We formulated a permutation problem on matrices dealing with the avoidance of certain forbidden submatrices. We investigated several special cases and surveyed related known results. There remains a number of (seemingly hard) open problems and questions that are spread across the paper; the number of open questions exceeds the number of known results by far. We think that we only scratched the surface of the problem and just took a first step towards a systematic investigation.

Apart from the open problems that are mentioned explicitly in the paper, there exist quite a lot of related questions that would deserve further work. In the following we shall mention just some of them.

(1) It is a challenging research question trying to generalize the investigations of this paper to multidimensional arrays. This could lead, for example, to a new class of greedily solvable linear programs. (Recently several authors successfully generalized the results mentioned in Section 2.7 on Monge matrices to the multidimensional case, see e.g., [6, 39].) Another interesting problem in this connection would be to search for possible areas of application of the multidimensional problem.

(2) What happens if we allow that the input matrix M contains some unspecified elements and we ask whether it is possible to fill these gaps such that M becomes a member of the set $\mathcal{M}(\mathcal{F})$ ($\mathcal{M}^c(\mathcal{F})$ or $\mathcal{M}^s(\mathcal{F})$) for a given set \mathcal{F} of forbidden matrices? (Somewhat related graph problems are treated in [25].)

(3) In this paper we mainly dealt with problems of type $\mathcal{M}(\mathcal{F})$ and $\mathcal{M}^c(\mathcal{F})$. Almost nothing seems to be known about the type $\mathcal{M}^s(\mathcal{F})$ where row and column permutations are required to be equal.

(4) We investigated only sets \mathcal{F} containing a single $(0, 1)$ -matrix of order 1×3 or 2×2 in a systematic way. A next step could be to treat $(0, 1)$ -matrices of order 1×4 or 2×3 . Likewise an in-depth treatment of 2×2 or perhaps even larger $(0, 2)$ -matrices along the lines of our study on $(0, 1)$ -matrices would be of interest. Particularly challenging seems to be to deal with forbidden sets \mathcal{F} with $|\mathcal{F}| \geq 2$.

Acknowledgement

We thank Alexander Ageev and Ron Shamir for helpful conversations and pointing out to us several results from the literature. Furthermore we are grateful to Ron Shamir, Jeremy Spinrad and an anonymous referee for their detailed comments on an earlier version of this paper which helped to improve the presentation.

References

- [1] I. Adler, A.J. Hoffman and R. Shamir, Monge and feasibility sequences in general flow problems, *Discrete Appl. Math.* 44 (1993) 21–38.
- [2] I. Adler and R. Shamir, Greedily solvable transportation networks and edge-guided vertex elimination, in: D.-Z. Du and P.M. Pardalos, eds., *Network Optimization Problems: Algorithms, Complexity and Applications* (World Scientific, Singapore, 1993) 1–22.
- [3] A.A. Ageev and V.L. Beresnev, Polynomially solvable special cases of the simple plant location problem, in: R. Kannan and W.R. Pulleyblank eds., *Proceedings of the First IPCO Conference* (Waterloo University Press, Waterloo, 1990) 1–6.
- [4] N. Alon, S. Cosares, D.S. Hochbaum and R. Shamir, An algorithm for the detection and construction of Monge sequences, *Linear Algebra Appl.* 114/115 (1989) 669–680.
- [5] R.P. Anstee and M. Farber, Characterizations of totally balanced matrices, *J. Algorithms* 5 (1984) 215–230.
- [6] W.W. Bein, P. Brucker, J.K. Park and P.K. Pathak, A Monge property for the d -dimensional transportation problem, *Discrete Appl. Math.* 58 (1995) 97–109.
- [7] V.L. Beresnev, Minimization problems for polynomials in Boolean variables, *Problemy Kibernet.* 36 (1979) 225–246 (in Russian)
- [8] V.L. Beresnev and A.I. Davydov, On matrices with connectedness property, *Upravlyayemye Sistemy* 19 (1979) 3–13 (in Russian).
- [9] C. Berge, Balanced matrices, *Math. Programming* 2 (1972) 19–31.
- [10] C. Berge, *Hypergraphs* (North-Holland, Amsterdam, 1989).
- [11] K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, *J. Comput. System Sci.* 13 (1976) 335–379.
- [12] R.E. Burkard, Remarks on some scheduling problems with algebraic objective function, *Methods Oper. Res.* 32 (1978) 63–77.
- [13] R.E. Burkard, On the role of bottleneck Monge matrices in combinatorial optimization, manuscript, Institute of Mathematics, University of Technology, Graz (1993); also: *Oper. Res. Lett.*, to appear.
- [14] L. Chen and Y. Yesha, Efficient parallel algorithms for bipartite permutation graphs, *Networks* 22 (1993) 29–39.
- [15] M. Conforti, G. Cornuéjols and M.R. Rao, Decomposition of balanced matrices, Part VII: A polynomial recognition algorithm, Management Science Research Report #MSRR-575, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA (1991).
- [16] M. Conforti and M.R. Rao, Structural properties and recognition of restricted and strongly unimodular matrices, *Math. Programming* 38 (1987) 17–27.
- [17] Y. Crama, P.L. Hammer and T. Ibaraki, Strong unimodularity for matrices and hypergraphs, *Discrete Appl. Math.* 15 (1986) 221–239.
- [18] M.T. de Berg and M.J. van Kreveld, Finding squares and rectangles in sets of points, *BIT* 31 (1991) 202–219.
- [19] M.T. de Berg, M. Overmars and M.J. van Kreveld, Finding complete bipartite subgraphs in bipartite graphs, *Algorithms Rev.* 1 (1990) 79–86.
- [20] V.G. Deineko and V.L. Filonenko, On the reconstruction of specially structured matrices, *Aktualnyye Problemy EVM, programmirovaniye, Dnepropetrovsk, DGU* (1979) (in Russian).
- [21] D.R. Fulkerson and O.A. Gross, Incidence matrices and interval graphs, *Pacific J. Math.* 15 (1965) 835–855.

- [22] H.N. Gabow and R.E. Tarjan, A linear-time algorithm for a special case of disjoint set union, *J. Comput. System Sci.* 30 (1985) 209–221.
- [23] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, CA, 1979).
- [24] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs* (Academic Press, New York, 1980).
- [25] M.C. Golumbic, H. Kaplan and R. Shamir, *Graphs sandwich problems*, Technical Report, The Moise and Frida Eskenasy Institute of Computer Sciences, Tel Aviv University (1992).
- [26] W. Gutjahr, E. Welzl and G.J. Woeginger, Polynomial graph-colorings, *Discrete Appl. Math.* 35 (1992) 29–45.
- [27] I.B.-A. Hartman, I. Newman and R. Ziv, On grid intersection graphs, *Discrete Math.* 87 (1991) 41–52.
- [28] A.J. Hoffman, On simple linear programming problems, in: V. Klee, ed., *Convexity, Proceedings of the Symposia in Pure Mathematics 7* (American Mathematical Society, Providence, RI, 1963) 317–327.
- [29] A.J. Hoffman, On greedy algorithms that succeed, in: *Surveys in Combinatorics*, London Mathematical Society Lecture Notes Series 103 (Cambridge University Press, Cambridge, 1985) 97–112.
- [30] A.J. Hoffman, A.W.J. Kolen and M. Sakarovitch, Totally-balanced and greedy matrices, *SIAM J. Algebraic Discrete Methods* 6 (1985) 721–730.
- [31] B. Klinz, R. Rudolf and G.J. Woeginger, On the recognition of bottleneck Monge matrices, Technical Report 241-92, Institute of Mathematics, University of Technology, Graz (1992); also: *Discrete Appl. Math.*, to appear.
- [32] J. Kratochvíl, A special planar satisfiability problem and a consequence of its NP-completeness, *Discrete Appl. Math.* 52 (1994) 233–252.
- [33] W. Lipski and F.P. Preparata, Efficient algorithms for finding maximum matchings in convex bipartite graphs and related problems, *Acta Inform.* 15 (1981) 329–346.
- [34] A. Lubiw, Doubly lexical orderings of matrices, *SIAM J. Comput.* 16 (1987) 854–879.
- [35] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, New York, 1988).
- [36] J. Nešetřil, Representation of graphs by means of products and their complexity, *Lecture Notes in Computer Science* 118 (Springer, Berlin, 1981) 94–102.
- [37] J. Opatrný, Total ordering problems, *SIAM J. Comput.* 8 (1979) 111–114.
- [38] R. Paige and R.E. Tarjan, Three partition refinement algorithms, *SIAM J. Comput.* 16 (1987) 973–989.
- [39] R. Rudolf, Recognition of d -dimensional Monge arrays, *Discrete Appl. Math.* 52 (1994) 71–82.
- [40] J. Spinrad, Doubly lexical ordering of dense 0-1 matrices, *Inform. Process. Lett.* 45 (1993) 229–235.
- [41] J. Spinrad, private communication.
- [42] J. Spinrad, A. Brandstädt and L. Stewart, Bipartite permutation graphs, *Discrete Appl. Math.* 18 (1987) 279–292.
- [43] A. Tucker, A structure theorem for the consecutive ones property, *J. Combin. Theory Ser. B* 12 (1972) 153–162.