

Perspective

 Perspectives of Monge properties in optimization [☆]

Rainer E. Burkard*, Bettina Klinz, Rüdiger Rudolf

TU Graz, Institut für Mathematik B, Steyrergasse 30, A-8010 Graz, Austria

Abstract

An $m \times n$ matrix C is called *Monge matrix* if $c_{ij} + c_{rs} \leq c_{is} + c_{rj}$ for all $1 \leq i < r \leq m$, $1 \leq j < s \leq n$. In this paper we present a survey on Monge matrices and related Monge properties and their role in combinatorial optimization. Specifically, we deal with the following three main topics: (i) fundamental combinatorial properties of Monge structures, (ii) applications of Monge properties to optimization problems and (iii) recognition of Monge properties.

Keywords: Monge property; Monge matrices; Combinatorial optimization; Recognition problems

1. Introduction and brief history

In this paper we study $m \times n$ matrices $C = (c_{ij})$ which fulfill the so-called Monge property

$$c_{ij} + c_{rs} \leq c_{is} + c_{rj} \quad \text{for all } 1 \leq i < r \leq m, 1 \leq j < s \leq n \quad (1)$$

or related properties obtained from (1) e.g. by replacing the composition “+” by other algebraic compositions or by relaxing the set of indices for which (1) has to hold.

Matrices with this property arise quite often in practical applications, especially in geometric settings. Take, e.g., a convex polygon and break it into two chains P and Q by removing two edges. Let P_1, \dots, P_m be the points on chain P ordered in clockwise order and let Q_1, \dots, Q_n be the points on chain Q ordered in counter-clockwise order. It is easy to check that the $m \times n$ matrix $C = (c_{ij})$ with entries $c_{ij} = d(P_i, Q_j)$, where $d(P_i, Q_j)$ denotes the Euclidean distance between the points P_i and Q_j , satisfies the Monge property (1). This follows from the quadrangle inequality for convex quadrangles. Other examples for Monge matrices can be found throughout the paper.

[☆] This research has been supported by the Spezialforschungsbereich F 003 “Optimierung und Kontrolle”, Projektbereich Diskrete Optimierung.

* Corresponding author. Tel. +43 316 873 5350; fax: +43 316 873 5369; e-mail: burkard@opt.math.tu-graz.ac.at

It is interesting to note that property (1) has a very long history. Already back in 1781 the French engineer and mathematician Gaspard Monge (1746–1818) considered the following problem in connection with transporting earth (cf. [96]):

Lorsqu'on doit transporter des terres d'un lieu dans un autre, on a coutume de donner le nom de Déblai au volume des terres que l'on doit transporter, et le nom de Remblai à l'espace qu'elles doivent occuper après le transport.

Le prix du transport d'une molécule étant, toutes choses d'ailleurs égales, proportionnel à son poids et à l'espace qu'on lui fait parcourir, et par conséquent le prix du transport total devant être proportionnel à la somme des produits des molécules multipliées chacune par l'espace parcouru, il s'ensuit que le déblai & le remblai étant donnés de figure et de position, il n'est pas indifférent que telle molécule du déblai soit transportée dans tel ou tel autre endroit du remblai, mais qu'il y a une certaine distribution à faire des molécules du premier dans le second, d'après laquelle la somme de ces produits sera la moindre possible, et le prix du transport total sera un minimum.....

Quelle que soit la route que doive suivre une molécule du déblai pour arriver au remblai, de manière que le prix du transport total soit un minimum, en supposant qu'il n'y ait point d'obstacle, cette route doit être une ligne droite; et en supposant qu'elle doive passer par des points déterminés, elle doit être une ligne droite d'un point à l'autre.

Lorsque le transport du déblai se fait de manière que la somme des produits des molécules par l'espace parcouru est un minimum, les routes de deux points quelconques A & B, ne doivent pas se couper entre leurs extrémités, car la somme Ab + Ba, des routes qui se coupent, est toujours plus grande que la somme Aa + Bb, de celles qui ne se coupent pas.

Je supposerai dans la suite que le déblai soit d'une densité uniforme, et qu'il soit divisé en molécules égales entre elles, afin que la question soit simplifiée & réduite à faire en forte que la somme des routes soit un minimum.

In other words, Monge wanted to split two equally large volumes (representing the initial location and the final location of the earth to be shipped) into infinitely many small particles and then associate them with each other so that the sum of the products of the lengths of the paths used by the particles and the volume of the particles is minimized. This problem can be viewed as special continuous transportation problem (see also Section 9).

For our purpose the most important observation of Monge is the following: If unit quantities have to be transported from locations P_1 and Q_1 to locations P_2 and Q_2 in such a way as to minimize the total distance travelled, then the route from P_1 and the route from Q_1 must not intersect (cf. Fig. 1). This follows again from the quadrangle inequality for convex quadrangles.

In 1961 Hoffman [75] rediscovered this observation of Monge and coined the term Monge property. Hoffman showed that the Hitchcock transportation problem can be solved by a very simple greedy approach if its underlying cost matrix satisfies property (1). (Actually, Hoffman considered a class of matrices which is larger than the

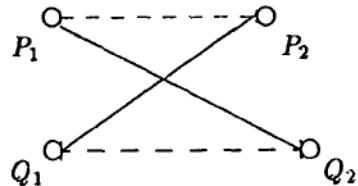


Fig. 1. Illustration of the Monge property

class of Monge matrices, namely the class of matrices which allow a Monge sequence, see Section 8 for further details.)

Until the mid-1980's Monge structures did not receive much attention in the combinatorial optimization or computer science community. But starting with the seminal paper by Aggarwal et al. [4] on searching in totally monotone matrices, a large number of researchers especially in theoretical computer science became interested in this field. Since then an enormous number of papers on applications which lead to Monge structures have been published in many different areas. As a consequence thereof, it became very difficult for a single researcher to keep track of even a small proportion of the recent papers on Monge properties and their applications. What makes this task even more difficult is that often the connection to Monge properties is not made explicit.

In this paper we give a survey on the rich literature on Monge properties and draw connections between results that seemed unrelated so far. Furthermore, we also report several new results. The main focus will be on the following three aspects: (i) fundamental combinatorial properties of Monge structures, (ii) applications of Monge structures in combinatorial optimization problems and (iii) recognition of Monge structures.

We are far from claiming to give a complete account of the work which has been done in connection with Monge properties. This has become simply impossible. In order to restrict the field to be covered by this survey at least to some extent, we concentrate on the role of Monge properties in combinatorial optimization problems and mention applications to other fields such as continuous optimization problems and computational geometry only very briefly. For similar reasons we will be concerned only with serial algorithms and do not deal with parallel algorithms. A good starting point for the reader interested in parallel algorithms in connection with Monge properties is the thesis of Park [102] and the references cited therein.

This paper is organized as follows: in Section 2 we introduce Monge matrices and investigate their combinatorial structure, while Section 3 lists a large number of applications which give rise to Monge matrices. Bottleneck Monge matrices which are obtained by replacing “+” in the Monge property (1) by “max” are discussed in Section 4. In Section 5 we demonstrate that both Monge matrices and bottleneck Monge matrices can be embedded into the larger framework of algebraic Monge matrices.

Section 6 deals with Monge properties in multidimensional arrays and their relevance to the field of combinatorial optimization. Hoffman's [75] notion of a Monge sequence and generalizations thereof are the topic of Section 8. Finally, in Section 9 various modifications of the Monge properties discussed in previous sections are considered. Among others we deal with a continuous Monge property, weak Monge matrices and Monge matrices with unspecified entries. The paper is closed with a short discussion and some concluding remarks in Section 10.

2. Monge matrices

2.1. Definitions

An $m \times n$ real matrix C is called a *Monge matrix* if C satisfies the so-called *Monge property*

$$c_{ij} + c_{rs} \leq c_{is} + c_{rj} \quad \text{for all } 1 \leq i < r \leq m, \quad 1 \leq j < s \leq n. \quad (2)$$

In some applications we need that the inequalities in (2) hold in the reverse direction. This motivates the following definition: An $m \times n$ real matrix C is called an *inverse Monge matrix* if C satisfies the *inverse Monge property*:

$$c_{ij} + c_{rs} \geq c_{is} + c_{rj} \quad \text{for all } 1 \leq i < r \leq m, \quad 1 \leq j < s \leq n. \quad (3)$$

Sometimes inverse Monge matrices are also called *contra Monge matrices* (see e.g. [128]) or *anti-Monge matrices* (see e.g. [33] or [67]). Note that an inverse Monge matrix can be transformed into a Monge matrix either by multiplying all entries by -1 or by reversing the order of its columns.

The above concept of Monge matrices can be slightly extended by allowing *infinite entries*. In this case the entries c_{ij} are drawn from the set $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ and the addition and the order \leq are extended to $\overline{\mathbb{R}}$ in the natural way, i.e. by requiring $a + \infty = \infty$ for all $a \in \overline{\mathbb{R}}$ and $a < \infty$ for all $a \in \mathbb{R}$. The entry ∞ can be used to model forbidden entries of the matrix which cannot be part of a feasible solution of the optimization problem under investigation.

Two important special cases of Monge matrices with infinite entries which arise very often in practice are *upper resp. lower triangular Monge matrices* which satisfy $c_{ij} = \infty$ for $i > j$ resp. $c_{ij} = \infty$ for $i < j$. Equivalently, C is an upper triangular Monge matrix iff

$$c_{ij} + c_{rs} \leq c_{is} + c_{rj} \quad \text{for all } 1 \leq i < r \leq j < s \leq n \quad (4)$$

and a lower triangular Monge matrix iff

$$c_{ij} + c_{rs} \leq c_{is} + c_{rj} \quad \text{for all } 1 \leq j < s \leq i < r \leq m. \quad (5)$$

Analogously, we can define lower and upper triangular inverse Monge matrices. Property (4) is also known in the literature as *concave quadrangle inequality* and the reversal of (4) is known as *convex quadrangle inequality* (cf. e.g. [136]).

2.2. Fundamental properties of Monge matrices

The class of Monge matrices has a lot of nice properties which play an important role in applications (cf. Section 3). Some of these properties will be summarized below. We mainly concentrate on Monge matrices, analogous results can easily be shown to hold for inverse Monge matrices as well.

We start with the straightforward observation that for real Monge matrices it suffices to require that the Monge property holds for adjacent rows and adjacent columns. In other words, condition (2) holds if and only if

$$c_{ij} + c_{i+1,j+1} \leq c_{i,j+1} + c_{i+1,j} \quad \text{for all } 1 \leq i < m, 1 \leq j < n. \quad (6)$$

An immediate consequence of this observation is that it can be tested in $O(nm)$ time whether a given $m \times n$ matrix is a Monge matrix. Note, however, that the equivalence above does not hold for Monge matrices with infinite entries (cf. also Section 5 for a theoretical explanation of this fact.)

An important subclass of Monge matrices can be generated in the following way: Let $D = (d_{ij})$ be a nonnegative real matrix of order $m \times n$. It is straightforward to prove that the matrix C obtained by

$$c_{ij} = \sum_{k=i}^m \sum_{\ell=1}^j d_{k\ell} \quad \text{for all } 1 \leq i \leq m, 1 \leq j \leq n \quad (7)$$

is a Monge matrix. In analogy to the notions of distribution and density matrices in probability theory, Gilmore et al. [66] call a matrix C which is given by (7) a *distribution matrix* generated by the *density matrix* D .

It has been observed by several authors (see e.g. [26, 23]) that Monge matrices are in fact only slightly more general than distribution matrices.

Lemma 2.1. *An $m \times n$ real matrix C is a Monge matrix if and only if there exists an $m \times n$ distribution matrix \widehat{C} and two vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$ such that*

$$c_{ij} = \widehat{c}_{ij} + u_i + v_j \quad \text{for all } 1 \leq i \leq m, 1 \leq j \leq n.$$

The following observation can be used to generate more complex (inverse) Monge matrices starting from very simple ones.

Observation 2.2. *Let C and D be two $m \times n$ (inverse) Monge matrices with entries from $\overline{\mathbb{R}}$ and let $u \in \overline{\mathbb{R}}^m$ and $v \in \overline{\mathbb{R}}^n$. Then the following matrices are (inverse) Monge matrices as well:*

- (i) the transpose C^T , (ii) the matrix λC for $\lambda \geq 0$, (iii) the sum $C + D$, (iv) the matrix $A = (a_{ij})$ defined by $a_{ij} := c_{ij} + u_i + v_j$.

Note that (ii) and (iii) imply that the class of (inverse) Monge matrices forms a convex cone K in the vector space of all $m \times n$ matrices with entries from $\overline{\mathbb{R}}$. Property (iv) is an immediate consequence of (iii).

In some applications it might be useful to know more about the cone K of Monge matrices. Obviously, K is a polyhedral cone and is thus finitely generated. But K is in general not a pointed cone since any *sum matrix*, i.e. any matrix $C = (c_{ij})$ of the form $c_{ij} = u_i + v_j$ and not only the zero matrix belongs to the intersection $K \cap -K$. The cone K^+ of all nonnegative Monge matrices is, however, pointed. In order to describe the extreme rays of K^+ we need the following classes of 0–1 matrices:

Let $H^{(p)}$ denote the 0–1 matrix whose p th row contains all ones while all other entries are zero and let $V^{(q)}$ denote the matrix whose q th column contains all ones while all other entries are zero. Obviously, the matrices $H^{(p)}$, $p = 1, \dots, m$, and $V^{(q)}$, $q = 1, \dots, n$, span the space of all sum matrices. Furthermore, let $R^{(pq)}$ be the 0–1 matrix with $r_{ij}^{(pq)} = 1$ for $i = 1, \dots, p$ and $j = q, \dots, n$ and $r_{ij}^{(pq)} = 0$ otherwise and let $L^{(pq)}$ be such that $\ell_{ij}^{(pq)} = 1$ for $i = p, \dots, m$ and $j = 1, \dots, q$ and $\ell_{ij}^{(pq)} = 0$ otherwise. Note that the matrices $R^{(pq)}$ are exactly those 0–1 distribution matrices which are generated by a density matrix which contains a single nonzero entry. The matrices $L^{(pq)}$ can also be seen as a kind of distribution matrices. We only have to exchange the role of rows and columns in formula (7).

The following characterization of the cone K^+ can be obtained from Lemma 2.1 and the properties of the basis matrices $H^{(p)}$, $V^{(q)}$, $R^{(pq)}$ and $L^{(pq)}$ mentioned above. A direct proof which does not use Lemma 2.1 can be found in Rudolf and Woeginger [113].

Lemma 2.3. *For any nonnegative $m \times n$ Monge matrix C there exist nonnegative numbers κ_i , λ_j , μ_{rs} and v_{pq} such that*

$$C = \sum_{i=1}^m \kappa_i H^{(i)} + \sum_{j=1}^n \lambda_j V^{(j)} + \sum_{r=2}^m \sum_{s=1}^{n-1} \mu_{rs} L^{(rs)} + \sum_{p=1}^{m-1} \sum_{q=2}^n v_{pq} R^{(pq)}. \quad (8)$$

The matrices $H^{(p)}$ with $p = 1, \dots, m$, $V^{(q)}$ with $q = 1, \dots, n$, $R^{(pq)}$ with $p = 1, \dots, m-1$, $q = 2, \dots, n$ and $L^{(pq)}$ with $p = 2, \dots, m$, $q = 1, \dots, n-1$ generate the extreme rays of the cone K^+ .

The matrices $L^{(pq)}$ are only needed to make sure that the numbers κ_i and λ_j are nonnegative. Note that by taking the transpose of the basis matrices, Lemma 2.3 immediately yields a characterization of the cone of all nonnegative inverse Monge matrices.

Rudolf and Woeginger [113] observed that Lemma 2.3 can be used to simplify optimality proofs for combinatorial optimization problems on (inverse) Monge matrices. In fact, whenever a combinatorial optimization problem has a solution which is optimal

for all basis matrices at the same time, this solution is also optimal for an arbitrary nonnegative Monge matrix. Further results on extreme rays for Monge-like matrices and applications thereof can be found in Burkard et al. [33], Çela, et al. [32] and in Girlich et al. [67].

While on the one hand it follows from the observations above that there are infinitely many Monge matrices, on the other hand the number of Monge matrices with small entries and distinct columns and distinct rows is rather small. More precisely, Klinz et al. [86] have shown that a 0–1 Monge matrix C with no identical rows and columns is of order at most 4×3 or 3×4 . This result carries over to Monge matrices with two different entries. Using similar arguments it can be shown that a Monge matrix with entries from the set $\{0, 1, \dots, k\}$ and distinct rows and distinct columns has at most $(k+1)^2$ rows and columns. (For $k \geq 2$ this does, however, not extend to matrices with $k+1$ different entries since in that case the cardinality of the set of differences of two matrix entries is in general much larger than for the case where the entries are drawn from the set $\{0, \dots, k\}$.)

Another property of (inverse) Monge matrices is fundamental for many applications. Let C be an $m \times n$ matrix and let $j(i)$ be the column of C which contains the leftmost minimum entry in row i , $1 \leq i \leq m$, and let $j'(i)$ be the column which contains the leftmost maximum entry in row i . We say that C is *monotone*, if

$$j(1) \leq j(2) \leq \dots \leq j(m) \quad (9)$$

or

$$j'(1) \leq j'(2) \leq \dots \leq j'(m) \quad (10)$$

holds. C is said to be *totally monotone* if all of its submatrices, or equivalently all 2×2 submatrices, are monotone. It is easy to see that C is totally monotone if and only if either

$$c_{ij} > c_{is} \implies c_{rj} > c_{rs} \quad \text{for all } 1 \leq i < r \leq m, 1 \leq j < s \leq n \quad (11)$$

or

$$c_{ij} < c_{is} \implies c_{rj} < c_{rs} \quad \text{for all } 1 \leq i < r \leq m, 1 \leq j < s \leq n. \quad (12)$$

It should be mentioned that some authors distinguish between matrices satisfying (11) and (12) (see e.g. [90]) while others use only one of the two properties to define totally monotone matrices (see e.g. [4]).

Based on the above definition of totally monotone matrices the following result can be shown easily.

Lemma 2.4. *Monge matrices and inverse Monge matrices are totally monotone matrices. Monge matrices fulfill (9) and (11) and inverse Monge matrices fulfill (10) and (12).*

Note that Lemma 2.4 remains true if we allow infinite entries.

2.3. Alternative characterizations of Monge matrices

In the following we deal with alternative characterizations of Monge matrices. We start with discussing the relationship between Monge matrices and submodular functions. To this end, consider a distributive lattice $\mathcal{L} = (\mathcal{I}, \vee, \wedge)$, where $\mathcal{I} = I_1 \times I_2 \times \cdots \times I_d$, $d \geq 2$, is a product space with $I_q \subseteq \mathbb{R}$, $q = 1, \dots, d$. The *join* and *meet* operations are defined as follows:

$$(x_1, \dots, x_d) \vee (y_1, \dots, y_d) := (\max\{x_1, y_1\}, \dots, \max\{x_d, y_d\}) \quad \text{for all } x, y \in \mathcal{I}$$

and

$$(x_1, \dots, x_d) \wedge (y_1, \dots, y_d) := (\min\{x_1, y_1\}, \dots, \min\{x_d, y_d\}) \quad \text{for all } x, y \in \mathcal{I}.$$

A function $f : \mathcal{I} \mapsto \mathbb{R}$ is said to be *submodular* (or *subadditive*) on \mathcal{L} if

$$f(x \vee y) + f(x \wedge y) \leq f(x) + f(y) \quad \text{for all } x, y \in \mathcal{I}. \quad (13)$$

If the inequality (13) holds in the other direction, then f is said to be *supermodular* (or *superadditive*).

Queyranne et al. [105] observed the following equivalence between Monge matrices and the class of submodular functions defined above.

Observation 2.5. Let $\mathcal{I} = \{1, \dots, m\} \times \{1, \dots, n\}$, $d = 2$ and define \vee and \wedge as above. An $m \times n$ matrix $C = (c_{ij})$ is a Monge matrix (resp. an inverse Monge matrix) if and only if the function $f : \mathcal{I} \mapsto \mathbb{R}$ defined by $f(i, j) := c_{ij}$ is submodular (resp. supermodular) on the lattice $\mathcal{L} = (\mathcal{I}, \vee, \wedge)$.

Consequently results that are known in the area of submodular functions can be translated into results for Monge matrices and vice versa.

In many cases it is, however, rather tedious to check whether or not a given function f from $\{1, \dots, m\} \times \{1, \dots, n\}$ onto the reals is submodular. The answer to this question becomes easier if f can be extended in a nice way to a function \tilde{f} defined on \mathbb{R}^2 .

First note that a function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ is submodular if and only if

$$f(x, y) + f(x', y') \leq f(x', y) + f(x, y') \quad \text{for all } x \leq x', y \leq y'. \quad (14)$$

Similarly, f is supermodular if and only if

$$f(x, y) + f(x', y') \geq f(x', y) + f(x, y') \quad \text{for all } x \leq x', y \leq y'. \quad (15)$$

Although properties (14) and (15) are nothing else than submodularity and supermodularity, respectively, they have been assigned many different names in the literature. Among others the terms *quasiantitone* and *2-antitone* have been used to refer to property (14) and the terms *quasimonotone*, *2-monotone* and *2-positive* for property (15) (cf. e.g. [106, 107] and the references given therein).

From (14) resp. (15) it is now easy to see that if the mixed second partial derivative $\partial^2 f / \partial x \partial y$ of f exists, f is submodular resp. supermodular on \mathbb{R}^2 if and only if $\partial^2 f / \partial x \partial y$ is ≤ 0 resp. ≥ 0 for all $x, y \in \mathbb{R}$.

Below we list several examples of simple functions f which satisfy (14) or (15):
(i) $f(x, y) := x + y$, (ii) $f(x, y) := \min\{x, y\}$, (iii) $f(x, y) := \max\{x, y\}$, (iv) $f(x, y) := x \cdot y$, (v) $f(x, y) := |x - y|^p$ for any $p \geq 1$, (vi) $f(x, y) := h(x - y)$ for any real function h which is either convex or concave, and (vii)

$$f(x, y) := \begin{cases} \int_x^y h_1(t) dt & x \leq y, \\ \int_y^x h_2(t) dt & y > x, \end{cases}$$

with integrable functions h_1 and h_2 such that $h_1(t) + h_2(t) \geq 0$ for all $t \geq 0$.

Furthermore, note that if f fulfills (14) (resp. (15)) and h_1 and h_2 are two real functions which are monotone in the same direction, then the function \tilde{f} defined by $\tilde{f}(x, y) := f(h_1(x), h_2(y))$ fulfills again (14) (resp. (15)). If h_1 and h_2 are monotone in opposite directions, then (14) turns into (15) and vice versa.

Many of the functions given above play an important role in connection with efficiently solvable special cases of the travelling salesman problem (see e.g. [66, 26, 127]).

3. Applications of Monge matrices

3.1. Uncapacitated transportation and flow problems

Given an $m \times n$ cost matrix C with entries from $\overline{\mathbb{R}}$, a nonnegative supply vector $a = (a_1, \dots, a_m)$ and a nonnegative demand vector $b = (b_1, \dots, b_n)$ such that $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$, the classical *Hitchcock transportation problem* (TP) can be formulated as follows:

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (16)$$

$$\text{s.t. } \sum_{j=1}^n x_{ij} = a_i \quad \text{for all } i = 1, \dots, m, \quad (17)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad \text{for all } j = 1, \dots, n, \quad (18)$$

$$x_{ij} \geq 0 \quad \text{for all } i, j. \quad (19)$$

Since the order of the demand nodes does not play a role in (TP), it is irrelevant whether the cost matrix C is a Monge matrix or an inverse Monge matrix. Hence

we can concentrate on Monge matrices, all results can be translated to inverse Monge matrices by reversing the order of the columns.

The theorem below follows directly from a more general result of Hoffman [75] on Monge sequences (cf. Section 8). Hoffman's result has been extended to the case of matrices with infinite entries by Shamir and Dietrich [117].

Theorem 3.1. *The north-west corner rule produces an optimal solution of (TP) for all supply and demand vectors a and b with $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ if and only if the cost matrix C is a Monge matrix.*

As a consequence of Theorem 3.1 (TP) can be solved in $O(n + m)$ time if the cost matrix C is (inverse) Monge, while the currently best algorithm for general cost matrices which is due to Orlin [100] runs in $O(n' \log n'(n' \log n' + m'))$ time where $n' = n + m$ and m' is the number of finite entries in C .

For the *linear assignment problem* (LAP), a special case of (TP) where $n = m$ and all supplies and demands are equal to one, the north-west corner rule always yields the identity permutation. Hence if C is an (inverse) Monge matrix, the corresponding (LAP) with cost matrix C has an optimal solution with a fixed structure which does not depend on C .

The results described above can be extended to uncapacitated minimum cost flow problems on nonbipartite graphs. Let $G = (V, E)$ be a directed graph with vertex set V , edge set E and m sources s_1, \dots, s_m and n sinks t_1, \dots, t_n and associate each edge $(i, j) \in E$ with a nonnegative cost c_{ij} . The basic observation due to Adler et al. [1] is that in the absence of capacities the flow between any source s_i and any sink t_j will be routed on a shortest path from s_i to t_j . Let d_{ij} be the length of the shortest path from source s_i to sink t_j w.r.t. the weights c_{ij} (set $d_{ij} := \infty$ if there is no such path at all). Obviously, the uncapacitated minimum cost flow problem can be transformed into a transportation problem with supply nodes s_1, \dots, s_m , demand nodes t_1, \dots, t_n and cost matrix $D = (d_{ij})$, and can hence be solved by a greedy approach based on the north-west corner rule if and only if the shortest distance matrix D is an (inverse) Monge matrix. The complexity of the resulting approach is dominated by the time needed for computing D . Since D can be computed by solving $O(\min\{n, m\})$ single-source shortest path problems, this algorithm is still faster than the best algorithm for the general uncapacitated minimum cost flow problem which requires $O(|V| \log |V|)$ shortest path computations (cf. [100]).

3.2. Unbalanced transportation problems

Unfortunately, Hoffman's greedy algorithm cannot be extended to *unbalanced transportation problems* where $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ and the equality sign in constraint (17) is replaced by \leq . The standard approach of introducing an artificial demand node and linking all supply nodes to this new node by zero-cost edges in general destroys the Monge property of the cost matrix.

A first step towards a fast algorithm for the unbalanced transportation problem on Monge matrices is made in Aggarwal et al. [3] where an $O(m \log(\sum_{i=1}^m a_i))$ time algorithm is given for the case of Monge matrices with *bitonic* rows, i.e. whenever each row consists of a monotone nonincreasing sequence that is followed by a monotone nondecreasing sequence, one of which may be empty.

Another special case for which a faster algorithm is known is a geometric instance of the unbalanced transportation problem which arises if all supply and demand points lie on a line and the transportation cost is given by the distance of the points along the line. For integer supplies and demands this problem can be solved in $O((n+m)\log(n+m))$ time as shown in Aggarwal et al. [3], who improved an older result of Karp and Li [84].

3.3. Minimax transportation problems

Let a nonnegative supply vector $a = (a_1, \dots, a_m)$ and a nonnegative demand vector $b = (b_1, \dots, b_n)$ such that $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ be given. Eiselt and Burkard [55] consider the following special case of the well-known minimax transportation problem:

$$\begin{aligned} \min \max \{x_{ij} : i = 1, \dots, m; j = 1, \dots, n\} \\ \text{s.t. (17) – (19).} \end{aligned} \quad (20)$$

Furthermore, they deal with the reshipment version of this problem where the constraints $x_{ij} \geq 0$ are dropped and the term x_{ij} in the objective function (20) is replaced by $|x_{ij}|$. For both problems they show that the optimal objective function value z^* can be obtained from an explicit formula which takes the form

$$z^* = \max \{h(p, q) : p = 1, \dots, m; q = 1, \dots, n\},$$

where $h(p, q)$ can be evaluated in constant time for any fixed p and q provided that a preprocessing is done, which takes $O(n + m)$ time. It can be shown that the matrix $C = (c_{ij})$ with entries $c_{ij} = h(i, j)$ is an inverse Monge matrix and thus is also totally monotone. This property is implicitly exploited by Eiselt and Burkard to obtain an $O(n + m)$ time algorithm for computing z^* and an $O((m + n)^2)$ time algorithm for computing an optimal transportation schedule (x_{ij}) . The latter algorithm also solves the lexicographic version of the problem where we ask for a solution (x_{ij}) such that the vector (\tilde{x}_{ij}) obtained from (x_{ij}) resp. from $(|x_{ij}|)$ by sorting the components nonincreasingly is minimal with respect to the usual lexicographic order.

3.4. Constrained transportation problems

Barnes and Hoffman [18] considered the transportation problem (TP) with the following additional constraints:

$$\sum_{r=1}^i \sum_{s=1}^j x_{rs} \leq d_{ij} \quad \text{for all } i = 1, \dots, m-1, j = 1, \dots, n-1 \quad (21)$$

where $D = (d_{ij})$ is a given $(m - 1) \times (n - 1)$ matrix. Suppose that the cost matrix C is a Monge matrix and that D is an inverse Monge matrix such that both the rows and columns of D are monotone nondecreasing. Barnes and Hoffman show that under these conditions a simple greedy approach yields a solution x_{ij} which is feasible and optimal, whenever a feasible solution exists at all. The proposed greedy approach is essentially the north-west corner rule with the exception that in maximizing the value of the current variable not only the constraints (17)–(19), but also constraint (21) has to be taken into consideration. Further generalizations of the cumulative capacity constraint (21) in connection with Monge matrices are discussed in Hoffman and Veinott [78].

A different type of constrained transportation problem is considered by Balinski and Rachev [17]. They show that if C is a Monge matrix, the problem

$$\begin{aligned} & \min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t. } & \sum_{i=1}^k \sum_{j=1}^n x_{ij} \leq \sum_{i=1}^k a_i = : \alpha_k \quad \text{for all } k = 1, \dots, m, \\ & \sum_{j=1}^{\ell} \sum_{i=1}^m x_{ij} \geq \sum_{j=1}^{\ell} b_j = : \beta_l \quad \text{for all } l = 1, \dots, n, \\ & x_{ij} \geq 0 \quad \text{for all } i, j \end{aligned}$$

always has an optimal solution (x_{ij}^*) of the following form: Let $r := \min\{p : a_p > 0\}$ and $s := \min\{q : b_q > 0\}$ and choose (i', j') such that $c_{i'j'} := \min\{c_{ij} : r \leq i \leq m, 1 \leq j \leq s\}$ (in case of ties take the lexicographically largest candidate pair). Then x^* can be chosen such that $x_{ij}^* = 0$ for all (i, j) lexicographically smaller than (i', j') and $x_{i'j'} = \min\{\sum_{p=1}^{i'} a_p, b_s\}$. This observation leads to a simple $O(nm)$ time greedy approach for solving the problem above.

A simpler greedy approach works for the special case where the cost matrix C is a symmetric square $n \times n$ Monge matrix with zeros on its main diagonal. Rachev and Rüschendorf [108] show that in that case the constrained transportation problem above can be transformed to an equivalent classical Hitchcock transportation problem with the same cost matrix C and new supplies \tilde{a}_i and demands \tilde{b}_j defined as follows: Let $\gamma_i := \max\{\alpha_i, \beta_i\}$, $1 \leq i \leq n$, $\delta_1 = \gamma_1$ and $\delta_{i+1} := \gamma_{i+1} - \gamma_i$, $1 \leq i \leq n - 1$. Then set $\tilde{a}_i := a_i$ and $\tilde{b}_i := \delta_i$ for all $1 \leq i \leq n$. Since the cost matrix C remains the same, the new transportation problem can be solved in $O(n)$ time by applying the north-west corner rule.

3.5. Further classes of greedily solvable linear programs

Bein et al. [19] study the concept of series and parallel compositions of linear programs and show that greedy properties are inherited by such compositions provided certain Monge and monotonicity properties are satisfied. Their work is inspired by

earlier results of Bein et al. [22] and Hoffman [77] on greedily solvable network flow problems on two-terminal series-parallel graphs, but drops the restriction that the underlying linear programs are specific descriptions of flow problems.

3.6. Weighted bipartite matching and k -factor problems

If capacity constraints of the form $x_{ij} \leq u_{ij}$, $i = 1, \dots, m$, $j = 1, \dots, n$, are added to the transportation models discussed above, the simple greedy approach does not work any longer. This remains true even for the special case $u_{ij} = 1$ for all i and j which is particularly important since the *weighted bipartite matching problem* and the *weighted bipartite k -factor problem* can be formulated in this framework. For the matching problem we set $a_i = b_j = 1$ for all i and j (if $n \neq m$ the resulting transportation problem is unbalanced), and for the k -factor problem we set $n = m$ and $a_i = b_j = k$ for all i and j . (Since in both cases the a_i and b_j are integral and the transportation constraints form a totally unimodular system, it is not necessary to require explicitly that the x_{ij} are integral.)

If $n = m$ holds in the weighted bipartite matching problem, the problem becomes equivalent to the linear assignment problem and can hence be solved in $O(n)$ time if C is (inverse) Monge. The best algorithm to solve the general unbalanced case $m > n$ is an $O((m - n + 1)n)$ time dynamic programming algorithm which is based on the simple observation that if C is Monge, there always exists a minimum weight matching which includes only matrix entries c_{ij} for which $1 \leq j - i \leq m - n + 1$. If all rows of the cost matrix C are, however, bitonic, then an improved $O(n \log m)$ time algorithm can be given (cf. [3]).

Similar ideas work for the minimum weighted bipartite k -factor problem which can be solved in $O((2k)^{4k}n)$ time by dynamic programming (see [104]). This algorithm exploits the fact that if C is Monge, there always exists an optimum k -factor which contains only matrix entries c_{ij} with $|i - j| < 2k$.

3.7. Searching and selecting in Monge matrices

The problem of determining the smallest or largest entries in certain rows or columns of a given matrix arises as subproblem in many applications and algorithms. Problems of this type can be solved considerably faster for (inverse) Monge matrices than for general matrices.

Let C be a given $m \times n$ Monge matrix. In the following we only discuss the problem of finding the minimum entry for each row of C . (The problem of finding all row maxima can be reduced to the row minima problem by multiplying all entries of C by -1 and reversing the order of the columns.) Aggarwal et al. [4] proposed a very efficient algorithm for computing all row minima of a totally monotone matrix fulfilling property (11). Below we briefly sketch the main ideas of this algorithm, the so-called SMAWK-algorithm.

Let again $j(i)$ denote the column which contains the leftmost minimum entry of row i , $i = 1, \dots, m$. Recall that Monge matrices fulfill $j(1) \leq \dots \leq j(m)$ (cf. (9)). We start with computing $j(\lceil \frac{m}{2} \rceil)$ which takes $O(n)$ time. Since $j(k) \leq j(\lceil \frac{m}{2} \rceil)$ for all $k < \lceil \frac{m}{2} \rceil$ and $j(k) \geq j(\lceil \frac{m}{2} \rceil)$ for all $k > \lceil \frac{m}{2} \rceil$, the original problem can be split into two smaller ones which are handled similarly. Applying this idea recursively leads to an algorithm which determines all row minima of an $m \times n$ Monge matrix in $O(n \log m)$ time.

In order to obtain an even faster algorithm, Aggarwal et al. [4] combine the above divide-and-conquer approach with two techniques which help in reducing the problem's size in each step. The first technique allows to eliminate rows of the currently investigated matrix and works as follows: Suppose the positions $j(i)$ for even numbered rows i are already known. Then due to (9) the remaining positions $j(i)$ for odd i can be computed in $O(n + m)$ time. The second more complicated technique aims at eliminating columns. In [4] it is shown that, given an $m \times n$ Monge matrix C with $m < n$, we can identify in $O(n)$ time $n - m$ columns of C that do not contain any leftmost row minimum. These columns can then be eliminated.

The combination of all ideas described above leads to an asymptotically optimal algorithm which finds all row minima of an $m \times n$ (inverse) Monge matrix in $O(n)$ time for $m \leq n$ and in $O(n(1 + \log(m/n)))$ time for $m > n$.

As an immediate consequence of the result discussed above it follows that the minimum entry and the maximum entry of an $m \times n$ (inverse) Monge matrix can be found in $O(n + m)$ time.

The above results on finding minima and maxima in (inverse) Monge matrices can be extended to selection problems. The k th smallest and the k -largest entry in each row of an (inverse) Monge matrix can be identified in $O(k(n+m))$ time using an algorithm of Kravets and Park [89] or in $O(n\sqrt{m \log n} + m)$ time due to Mansour et al. [93]. The k th smallest (largest) entry overall can be found in $O(n + m + k \log(nm/k))$ time (cf. [89]) or in $O((m+n)\sqrt{n} \log n)$ time (cf. [11]).

3.8. Dynamic programming

Dynamic programming is an important paradigm for solving many classes of optimization problems. It turns out that quite often dynamic programming algorithms can be speeded up if the underlying weights satisfy the (inverse) Monge property.

The first result in this direction is due to Wilber [133]. Let weights w_{ij} , $1 \leq i < j \leq n$, be given such that the matrix $W = (w_{ij})$ is an $n \times n$ upper triangular Monge matrix with $w_{ij} = \infty$ for $i \geq j$. Here and in the following it suffices if each w_{ij} can be computed in constant time. It is not necessary that the weights are given explicitly. Wilber considered the following dynamic programming setting:

$$d_{ij} := \begin{cases} E(i) + w_{ij} & \text{if } i < j, \\ \infty & \text{otherwise,} \end{cases} \quad (22)$$

where $E(1)$ is given and $E(i)$, $i = 2, \dots, n$, can be computed in constant time from the minimum entry in column i of matrix $D = (d_{ij})$. The aim is to determine the values

$E(2), \dots, E(n)$. Note that the value of $E(i)$ can only be computed after the i th column of D has been computed. Hence the problem of computing the values $E(i)$ can be regarded as *on-line* version of the column minimization problem discussed above.

Recurrences of the type (22) arise in many applications, e.g. in connection with the concave least-weight subsequence problem (cf. [73, 133]), special cases of the travelling salesman problem and of economic lot-sizing problems (see [103, 8]). The last two applications will be discussed in further detail below.

It can be checked that if W is Monge, then also D is Monge. Since solving (22) is an on-line problem, the off-line SMAWK-algorithm of Aggarwal et al. [4] cannot be applied. Wilber developed, however, an extension of the SMAWK-algorithm to the on-line case and obtained an $O(n)$ time algorithm for computing all $E(i)$. A disadvantage of Wilber's algorithm is that it guesses a block of values $E(i)$ at once, then computes the corresponding values of d_{ij} and checks in the end of each iteration if the guessed values are correct. If this is not the case, the incorrect values and several values of d_{ij} have to be recomputed. This property of Wilber's approach causes problems if we require that the values $E(i)$ have to be computed in order, i.e. $E(j)$ can only be evaluated when all the values $E(1), \dots, E(j-1)$ are known before.

Eppstein [56] presented a modification of Wilber's algorithm which circumvents this problems and still runs in linear time. This result is significant in that it allows the computation of the values $E(1), \dots, E(n)$ to be interleaved with the computation of another sequence $F(1), \dots, F(n)$ such that $E(j)$ depends on $F(1), \dots, F(j-1)$ and $F(j)$ depends on $E(1), \dots, E(j)$. For even further generalizations and extensions the interested reader is referred to Galil and Park [65], Larmore and Schieber [90] and the references given therein.

Wilber's approach works only if the weight matrix W is a Monge matrix. The case of inverse Monge matrices seems to be more difficult. What causes problems in that case is that the matrix D defined in (22) is not an inverse Monge matrix and does not fulfill the conditions which are required to apply the SMAWK-algorithm. Still, however, the problem can be solved more efficiently than for arbitrary weights w_{ij} , as is shown in Klawe and Kleitman [85] where an $O(n\alpha(n))$ time algorithm for computing all $E(i)$ is given ($\alpha(n)$ denotes the inverse Ackermann function).

Larmore and Schieber [90] considered the following two-dimensional dynamic programming recurrence: Let weights w_{ij} , $0 \leq i < j \leq 2n$, be given and define

$$D(i, j) := \min \{F(i', j') + w_{i'+j', i+j} : 0 \leq i' < i, 0 \leq j' < j\} \quad \text{for all } 1 \leq i, j \leq n, \quad (23)$$

where it is assumed that the values $F(i, 0)$ and $F(0, i)$ are given for $0 \leq i \leq n$ and that the remaining values of F can be computed in constant time from the corresponding entries of D . That is, $F(i, j)$ is available only after $D(i, j)$ has been calculated. This problem arises in the prediction of RNA secondary structure from the primary (linear) RNA structure and is known as Waterman's problem (for further details cf. [132]).

Relying on the results for the one-dimensional recurrence (22) discussed above, Laromore and Schieber show that Waterman's problem can be solved in optimal $O(n^2)$ time if the weights w_{ij} satisfy the concave quadrangle inequality (4) (i.e. if the matrix \tilde{W} with $\tilde{w}_{ij} = w_{ij}$ for $i < j$ and $\tilde{w}_{ij} = \infty$ for $i \geq j$ is an upper triangular Monge matrix). The case of weights satisfying the convex quadrangle inequality seems again to be more difficult. The fastest known algorithm runs in $O(n^2\alpha(n))$ time (cf. [90]).

A third type of dynamic programming recurrence that can be solved faster if the underlying weights are (inverse) Monge is treated in Yao [135, 136] (see also Park [102] who put Yao's work into a Monge framework). Let an $n \times n$ weight matrix $W = (w_{ij})$ with $w_{ij} = \infty$ for $i > j$ be given. Yao studies recursions of the form

$$D(i,j) := \begin{cases} 0 & \text{if } i = j, \\ w_{ij} + \min_{i \leq q < j} \{D(i,q) + D(q+1,j)\} & \text{if } i < j, \end{cases} \quad (24)$$

for $1 \leq i \leq j \leq n$. Yao proved that if W is (upper triangular) Monge, i.e. the w_{ij} , $1 \leq i \leq j \leq n$, satisfy (4), and if furthermore the following monotonicity constraint:

$$w_{ij} \leq w_{rs} \quad \text{for all } 1 \leq r \leq i \leq j \leq s \leq n \quad (25)$$

holds, then the recurrence (24) can be solved in $O(n^2)$ time. Therein property (25) guarantees that the Monge property of W is inherited to the matrix $C = (c_{ij})$ defined by $c_{ij} = D(i,j)$ for $i \leq j$ and $c_{ij} = \infty$ for $i > j$.

For other types of recursions different from (24) which satisfy Yao's conditions and can hence be solved faster than for general weights see Borchers and Gupta [25]. Another extension is due to Aggarwal and Park [5, 6, 102], who showed that if the weight matrix W is (upper triangular) inverse Monge, and if the inequality in (25) holds in the other direction, then (24) can be solved in $O(n^2\alpha(n))$ time.

3.9. Travelling salesman problems

Given n cities and an $n \times n$ cost matrix C , the travelling salesman problem (TSP) asks for a *tour* of minimum overall cost. Equivalently, the (TSP) can be formulated as follows:

$$\min_{\phi} \sum_{i=1}^n c_{i\phi(i)} : \quad \phi \text{ is a cyclic permutation.}$$

Since the (TSP) is NP-hard in general, polynomially solvable special cases are of great importance. In the following we report on well-solved special cases of the (TSP) which are related to Monge matrices.

Efficiently solvable special cases of the (TSP) almost go hand in hand with the concept of *pyramidal tours*: A tour $\phi = \langle 1, i_1, i_2, \dots, i_r, n, j_1, \dots, j_{n-r-2} \rangle$ is called *pyramidal*, if $i_1 < i_2 < \dots < i_r$ and $j_1 > \dots > j_{n-r-2}$.

The following theorem of Gilmore et al. [66] relates pyramidal tours and Monge matrices.

Theorem 3.2. *If the cost matrix C of a (TSP) is a Monge matrix, then there exists an optimal tour which is pyramidal.*

The class of pyramidal tours is interesting because despite the fact that the number of pyramidal tours is exponential in the number of cities, a minimum cost pyramidal tour can be determined in $O(n^2)$ time by dynamic programming (cf. [66]).

Let $D(i,j)$ denote the length of a minimum cost Hamiltonian path from i to j which visits all cities in $1, 2, \dots, \max\{i, j\}$ and passes through the cities in descending order of index from i to 1 and then through the complementary set of cities in ascending order of index from 1 to j . A path fulfilling these restrictions will, in the following, be referred to as *pyramidal path*. Gilmore et al. gave recursive formulae to compute all $D(i,j)$ and also an optimal tour in $O(n^2)$ time. In the following we present a slightly different recursion due to Park [103] which turns out to be advantageous if C is a Monge matrix.

Park computes only the values $D(i,j)$ for $|i-j|=1$. Observing that a minimum cost pyramidal path from j to $j+1$ passing through the cities $1, \dots, j+1$ can be decomposed into three parts, namely (i) an edge $(i, j+1)$ such that $1 \leq i < j$, (ii) a path from vertex j to vertex $i+1$ passing through the cities $i+1, \dots, j$ in strictly descending order, and (iii) a minimum cost pyramidal path from vertex $i+1$ to i passing through the cities $1, \dots, i+1$, we get

$$D(j, j+1) = \min_{i < j} \{D(i+1, i) + c_{i, j+1} + \sum_{\ell=i+1}^{j-1} c_{\ell+1, \ell}\} \quad \text{for } 2 \leq j \leq n-1. \quad (26)$$

By a similar argument we obtain

$$D(j+1, j) = \min_{i < j} \{D(i, i+1) + c_{j+1, i} + \sum_{\ell=i+1}^{j-1} c_{\ell, \ell+1}\} \quad \text{for } 2 \leq j \leq n-1. \quad (27)$$

Obviously, we have $D(1, 2) = c_{12}$ and $D(2, 1) = c_{21}$. The cost of an optimal tour is given by

$$\min\{D(n-1, n) + c_{n, n-1}, D(n, n-1) + c_{n-1, n}\}.$$

Park showed that the recurrences (26) and (27) can be solved in $O(n)$ time if C is a Monge matrix. He proceeds as follows: Let $F(j) := D(j, j+1)$ and $G(j) := D(j+1, j)$ and introduce the following $(n-1) \times (n-1)$ matrices A and B :

$$a_{ij} := \begin{cases} G(i) + c_{i, j+1} + \sum_{\ell=i+1}^{j-1} c_{\ell+1, \ell} & \text{if } i < j, \\ \infty & \text{if } i \geq j \end{cases}$$

and

$$b_{ij} := \begin{cases} F(i) + c_{j+1,i} + \sum_{\ell=i+1}^{j-1} c_{\ell,\ell+1} & \text{if } i < j, \\ \infty & \text{if } i \geq j. \end{cases}$$

Clearly, $F(j)$ and $G(j)$ are the minimum entries of the j th column of A and B , respectively. Park [103] proved that if C is Monge, A and B are Monge matrices as well. Thus, the algorithm of Eppstein [56] can be applied to compute the values $F(j)$ and $G(j)$, $j = 1, \dots, n$, and an optimal tour in $O(n)$ time.

The situation becomes more difficult if the cost matrix C is not Monge, but inverse Monge. A direct approach to the (TSP) on inverse Monge matrices is due to Aizenshtat and Kravchuk [12] and Aizenshtat and Maksimovich [13], who gave a rather complicated characterization of the structure of an optimal tour and derived an $O(n^2)$ time algorithm for finding an optimal tour.

In the following we briefly sketch the main ideas of a more general approach which leads to a linear time algorithm for the (TSP) on inverse Monge matrices. Let τ be the optimal solution of the linear assignment problem with cost matrix C and let τ_1, \dots, τ_q be the cycles in the cyclic representation of τ . If $q = 1$, we are obviously finished, if $q > 1$ the so-called *patching graph* $G_\tau = (V_\tau, E_\tau)$ associated with τ plays an important role. In G_τ there is a vertex v_j for each cycle τ_j for $j = 1, \dots, q$ and an edge (v_j, v_s) which is labelled by $(i, i+1)$ if and only if index i occurs in cycle τ_j and index $(i+1)$ occurs in cycle τ_s . G_τ is a connected multigraph with at most $n - 1$ edges (cf. [66]). Then the following holds (see e.g. [64, 66]):

Theorem 3.3. *Let τ be an optimal solution of the linear assignment problem with cost matrix C . If $C_\tau = (c_{i\tau(j)})$ is a Monge matrix and if the patching graph G_τ is a multipath, then the travelling salesman problem with cost matrix C can be solved in polynomial time.*

Sarvanov [114] and Deĭneko [46] independently obtained an $O(n^3)$ algorithm to solve the special case of the (TSP) described in Theorem 3.3. Gaikov [64] reduced the running time of Sarvanov's algorithm down to $O(nq)$. Finally, Burkard and Deĭneko [34] achieved a further improvement. Their algorithm, which relies on ideas from [103], runs in $O(n)$ time provided that the optimal assignment τ is already known.

Since for inverse Monge matrices we can choose τ such that $\tau(i) = n + 1 - i$ for $i = 1, \dots, n$, which leads to a multipath G_τ (there are exactly two edges between the nodes v_j and v_{j+1} , $1 \leq j < \lfloor (n+1)/2 \rfloor$), the assumptions of Theorem 3.3 are obviously fulfilled. Thus the (TSP) on inverse Monge can be solved in $O(n)$ time.

Further results on the travelling salesman problem in connection with Monge and Monge-like matrices and patching graphs can be found in Gaikov [64] and in Burkard et al. [36]. For a survey of results of this type and many other results on special cases of the (TSP) the reader is also referred to the survey article by Burkard et al. [35].

3.10. Economic lot-sizing problems

The basic *economic lot-sizing problem* which arises in production/inventory systems can be described as follows: Let the production horizon be divided into n production periods and let d_i be the demand that occurs during period i . This demand can be satisfied from the production x_i during period i or from the inventory which remained from the production during earlier periods. Further, let $c_i(x)$ denote the cost of producing x units of inventory during period i and let $h_i(y)$ be the cost of storing y units of inventory from period $i - 1$ to period i , $i = 1, \dots, n$. If we furthermore require that the initial and the final inventory are zero, then a feasible production schedule (x_1, \dots, x_n) can be characterized by the following system of constraints:

$$x_i + y_i - y_{i+1} = d_i \quad \text{for } i = 1, \dots, n,$$

$$y_1 = y_{n+1} = 0 \quad \text{and} \quad x_i, y_i \geq 0 \quad \text{for } i = 1, \dots, n,$$

where y_i represents the number of units that are kept over from period $i - 1$ to period i .

The basic economic lot-sizing problem is to find a feasible production schedule which minimizes the total induced cost given by

$$\sum_{i=1}^n c_i(x_i) + \sum_{i=2}^n h_i(y_i).$$

It is well known that this problem is NP-hard in general. If all c_i and h_i are concave, the problem can, however, as observed by Veinott [129] be solved in $O(n^2)$ time by a dynamic programming approach first proposed by Wagner and Whitin [131]. In the following we will only be concerned with this special case.

Let $E(j)$ denote the minimum cost of a feasible production schedule for periods 1 through $j - 1$ such that the inventory y_j carried forward to period j is zero. Since there always exists a minimum cost production schedule such that the demand in each period j is either fully supplied from the production in period j or from the inventory kept over from period $j - 1$, we obtain the following recurrence:

$$E(j) = \min_{1 \leq i < j} \left\{ E(i) + c_i(d_{ij}) + \sum_{q=i+1}^{j-1} h_q(d_{qj}) \right\},$$

where $d_{ij} = \sum_{q=i}^{j-1} d_q$ for $1 \leq i < j \leq n + 1$. Obviously, $E(n + 1)$ gives the cost of an optimal production schedule for the original n period problem.

Aggarwal and Park [8] deal with two special cases of the above problem. In both cases the storage costs are linear, say $h_i(y) = h_i^1 y$, and the production costs are of fixed cost type, i.e. $c_i(x) = c_i^0 + c_i^1 x$ for $x > 0$ and $c_i(0) = 0$ with $c_i^0 \geq 0$. In the first case

it is additionally assumed that $c_i^1 \leq c_{i-1}^1 + h_i^1$ for $i = 2, \dots, n$ which implies that the $n \times (n+1)$ matrix A with entries $a_{ij} = E(i) + c_i^0 + c_i^1 d_{ij} + \sum_{q=i+1}^{j-1} h_q^1 d_{qj}$ for $i < j$ and $a_{ij} = \infty$ otherwise, is a Monge matrix. Since we have

$$E(j) = \begin{cases} E(j-1) & \text{if } d_{j-1} = 0 \\ \min_{1 \leq i < n} a_{ij} & \text{if } d_{j-1} > 0, \end{cases}$$

the on-line matrix searching techniques described above yield an $O(n)$ time algorithm for this special case of the economic lot-sizing problem. For the case of arbitrary coefficients c_i^1 and h_i^1 , Aggarwal and Park [8] give an $O(n \log n)$ time algorithm which combines a divide-and-conquer approach with the above dynamic programming setting. The basic idea is to determine a permutation π of $\{1, \dots, n\}$ such that $c_{\pi(1)}^1 - \sum_{q=1}^1 h_{\pi(q)}^1 \geq c_{\pi(2)}^1 - \sum_{q=1}^2 h_{\pi(q)}^1 \geq \dots \geq c_{\pi(n)}^1 - \sum_{q=1}^n h_{\pi(q)}^1$. Reordering the rows of the matrix A according to π results in a Monge-like matrix. Further details can be found in [8].

Aggarwal and Park [8] also extend their techniques to the economic lot-sizing problem with *backlogging*. In this variation of the problem the demand may remain unsatisfied during some period, provided that it is satisfied eventually by production in some subsequent period. This means that in the mathematical formulation of the problem the constraints $y_2, \dots, y_n \geq 0$ may be dropped. For $y_i < 0$ the value $h_i(y_i)$, $y_i < 0$, now gives the cost of having a shortage of y_i units at the start of period i . For the sake of clarity, define backlogging cost functions g_i such that $g_{i-1}(-y_i) := h_i(y_i)$ for $y_i < 0$, $i = 2, \dots, n$.

Zangwill [137] showed that the economic lot-sizing problem with backlogging can be solved in $O(n^3)$ time if all cost functions c_i , h_i and g_i are concave on $[0, \infty)$. Using Monge techniques and the SMAWK off-line matrix searching algorithm, Aggarwal and Park [8] obtain an improved $O(n^2)$ time algorithm for this problem. Furthermore, they derive an $O(n \log n)$ time algorithm for the special case where $h_i(y) = h_i^1 y$, $g_i(z) = g_i^1 z$ and $c_i(x) = c_i^0 + c_i^1 x$ with $c_i^0 \geq 0$ for $x > 0$ and $c_i(0) = 0$ and an $O(n)$ time algorithm under the additional assumption $c_i^1 \leq c_{i-1}^1 + h_i^1$ for $i = 2, \dots, n$ and $c_i^1 \leq c_{i+1}^1 + g_i^1$ for $i = 1, \dots, n-1$.

These last two results as well as the results for the problem without backlogging discussed above have independently been obtained also by Federgruen and Tzur [60] and Wagelmans et al. [130] using substantially different techniques. The Monge techniques of Aggarwal and Park seem, however, to be more powerful as they yield faster algorithms for the general concave problem treated in Zangwill [137] as well as for several other types of economic lot-sizing problems with periodic data discussed in Graves and Orlin [70] and Erickson, et al. [57] (for details the reader is referred to Aggarwal and Park [8]). Furthermore, as shown in Aggarwal and Park [9], Monge techniques can also be applied to special cases of capacitated economic lot-sizing problems with bounds on production, inventory and backlogging.

3.11. Geometric applications

The Monge property and the quadrangle inequality for convex quadrangles are strongly related. Thus it is not surprising that Monge matrices arise in many geometric problems. In fact, all the examples discussed in the article of Aggarwal et al. [4] on searching in totally monotone arrays give rise to (inverse) Monge matrices. Since the main focus of this survey is on the Monge property in connection with combinatorial optimization problems, we will only give a few illustrative examples of applications of Monge matrices in geometric problems. Further details and a large number of additional references can be found in Aggarwal et al. [4] and Park [102].

Let a convex polygon P with vertices v_1, \dots, v_n in clockwise order be given. Among others the following results can be obtained by applying Monge techniques: (i) The farthest/nearest neighbour v_j for each vertex v_i , $i = 1, \dots, n$, can be determined in $O(n)$ time overall (cf. [4]). (ii) For given d , $3 \leq d \leq n$, the maximum-perimeter or maximum-area inscribed d -gon of P can be found in $O(dn + n \log n)$ time (cf. [4]). (iii) If n is even, a minimum weight perfect matching of the vertices of P where the weight of the edge linking v_i and v_j is the Euclidean distance of v_i and v_j can be computed in $O(n \log n)$ time (cf. [94]).

3.12. Path problems

Let $G = (V, E)$ be a directed graph with vertex set $V = \{1, \dots, n\}$ and edge set E where a real weight w_{ij} is attached to each edge $(i, j) \in E$. G is called an (*inverse*) *Monge graph* if the matrix $C = (c_{ij})$ defined by $c_{ij} = w_{ij}$ for $(i, j) \in E$ and $c_{ij} = \infty$ for $(i, j) \notin E$ is an (*inverse*) Monge matrix.

The following observation is well-known and can be proved in the same way as Theorem 3.2 for the travelling salesman problem.

Lemma 3.4. *In a Monge graph G there always exists a shortest path P from vertex 1 to vertex n which visits the vertices in ascending order, i.e. $P = (i_0, i_1, \dots, i_\ell, i_{\ell+1})$ with $i_0 = 1 < i_1 < \dots < i_\ell < i_{\ell+1} = n$.*

The above lemma implies that setting $w_{ij} = \infty$ for all $i \geq j$ has no effect on the shortest path from 1 to n . We may therefore assume that G is acyclic and that $(i, j) \in E$ implies $i < j$. Since a shortest path in an acyclic graph can be determined by a dynamic programming recurrence of the type described in (22), Wilber's result in [133] yields an $O(n)$ time algorithm for computing a shortest 1 to n path in a Monge graph G .

A similar idea can be applied to the problem of computing a shortest path from 1 to n with exactly k edges. The first algorithm for this problem is due to Aggarwal et al. [4]. It combines a simple dynamic programming approach with k calls to the SMAWK-algorithm for matrix searching and therefore runs in $O(kn)$ time. Using a parametric reformulation of the problem and a refined search technique Aggarwal

et al. [10] were able to obtain an $O(n\sqrt{k \log n})$ time algorithm which improves on the naive algorithm for large values of k . A further improvement was obtained by Schieber [115], who derived an $O(n2^{O(\sqrt{\log k \log \log n})})$ time algorithm.

The corresponding path problems on inverse Monge graphs are more difficult to solve. One of the reasons is that Lemma 3.4 does not hold. Thus we need to require explicitly that the graph G is acyclic and that the nodes are numbered such that $(i, j) \in E$ implies $i < j$. Under this additional assumption a shortest path from 1 to n in an inverse Monge graph G can be found in $O(n\alpha(n))$ time by applying the algorithm of Klawe and Kleitmann [85]. The best algorithm for computing a shortest 1 to n path with exactly k edges is still the naive dynamic programming algorithm which runs in $O(nk\alpha(n))$ time and relies on repeated applications of the algorithm of Klawe and Kleitman [85].

Now let $G = (V, W; E)$ denote an undirected bipartite graph with vertex set $V \cup W$, where $V = \{v_1, \dots, v_m\}$ and $W = \{w_1, \dots, w_n\}$. Each edge $(i, j) \in E$ is attached a weight c_{ij} . Pferschy et al. [104] showed that if the matrix $C = (c_{ij})$ is a Monge matrix, a longest path from node v_1 to node v_m can be found in $O(nm)$ time by a dynamic programming approach which exploits the special structure of an optimal path.

4. Bottleneck Monge matrices

4.1. Definitions and fundamental properties of bottleneck Monge matrices

Replacing “+” by “max” in the Monge property (2) we arrive at the so-called *bottleneck Monge property*:

$$\max\{c_{ij}, c_{rs}\} \leq \max\{c_{ir}, c_{js}\} \quad \text{for all } 1 \leq i < r \leq m, 1 \leq j < s \leq n. \quad (28)$$

An $m \times n$ matrix C is called *bottleneck Monge matrix* if it fulfills the bottleneck Monge property. *Inverse bottleneck Monge matrices* and *bottleneck Monge matrices with infinite entries* are defined analogously as in the sum case. In the case of bottleneck Monge matrices infinite entries are, however, not necessary since they can always be replaced by a sufficiently large number (larger than all remaining entries) without destroying the bottleneck Monge property. This is not true for Monge matrices.

It is easy to see that there exist Monge matrices which are not bottleneck Monge and bottleneck Monge matrices which are not Monge. Note, however, that any 0–1 Monge matrix is also a bottleneck Monge matrix.

At first sight one might hope that bottleneck Monge matrices and Monge matrices have similar properties. Unfortunately, this is not true in general. While it suffices to require the Monge property for adjacent rows and adjacent columns, the corresponding property

$$\max\{c_{i,j}, c_{i+1,j+1}\} \leq \max\{c_{i+1,j}, c_{i,j+1}\} \quad \text{for all } 1 \leq i < m, 1 \leq j < n \quad (29)$$

is not equivalent to the bottleneck Monge property as can be seen from the following example matrix which satisfies (29), but is not bottleneck Monge:

$$C = \begin{pmatrix} 4 & 7 & 1 \\ 3 & 7 & 2 \end{pmatrix}.$$

Another difference between the two classes is that bottleneck Monge matrices are in general not totally monotone which has far-reaching algorithmic consequences as will turn out later on.

Below we list several properties that Monge matrices and bottleneck Monge matrices have in common. Obviously, the statements in Observation 2.2 can be carried over directly to the bottleneck case by replacing the componentwise addition in (iii) and (iv) by the componentwise maximum operation.

In order to generalize Lemma 2.1 we need to introduce the following special class of bottleneck Monge matrices: Given a nonnegative matrix D , the matrix C obtained by setting

$$c_{ij} = \max\{d_{k\ell} : i \leq k \leq m, 1 \leq \ell \leq j\} \quad \text{for all } 1 \leq i \leq m, 1 \leq j \leq n \quad (30)$$

or

$$c_{ij} = \max\{d_{k\ell} : 1 \leq k \leq i, j \leq \ell \leq n\} \quad \text{for all } 1 \leq i \leq m, 1 \leq j \leq n \quad (31)$$

is called a *max-distribution matrix*.

Lemma 4.1. *An $m \times n$ nonnegative matrix C is a bottleneck Monge matrix if and only if there exists an $m \times n$ max-distribution matrix \hat{C} and two vectors $u \in \mathbb{R}_+^m$ and $v \in \mathbb{R}_+^n$ such that*

$$c_{ij} = \max\{\hat{c}_{ij}, u_i, v_j\} \quad \text{for all } 1 \leq i \leq m, 1 \leq j \leq n.$$

Note that this statement becomes wrong if the class of max-distribution matrices is defined only on the basis of (30). Since in bottleneck Monge matrices only the relative order of the entries but not their actual size plays a role, it causes no loss of generality that Lemma 4.1 is formulated only for nonnegative matrices. Lemma 4.1 can be used to obtain an analogue of Lemma 2.3: just replace every occurrence of “+” by “max”.

Since bottleneck Monge matrices are in general not totally monotone, but this property is very important for many applications, Bein et al. [20] introduced a stronger property which leads to a totally monotone subclass of bottleneck Monge matrices. They call an $m \times n$ matrix C a *strict bottleneck Monge matrix* if for all $1 \leq i < r \leq m$, $1 \leq j < s \leq n$ one of the following two properties is fulfilled:

- (i) $\max\{c_{ij}, c_{rs}\} < \max\{c_{is}, c_{rj}\}$ or (32)
- (ii) $\max\{c_{ij}, c_{rs}\} = \max\{c_{is}, c_{rj}\}$ and $\min\{c_{ij}, c_{rs}\} \leq \min\{c_{is}, c_{rj}\}$.

Note that in particular matrices with pairwise distinct entries fulfill property (32).

Strict bottleneck Monge matrices and Monge matrices are closely related. More specifically, given a strict bottleneck Monge matrix C , there always exists an integer $\alpha > 1$ such that the matrix $B = (b_{ij})$ obtained by setting $b_{ij} := \alpha^{c_{ij}}$ is a Monge matrix. This transformation does not work for bottleneck Monge matrices in general, take, e.g. the matrix

$$C = \begin{pmatrix} 3 & 7 & 1 \\ 4 & 7 & 2 \end{pmatrix}.$$

At first sight one might hope that one can get around this problem by applying a small perturbation to the elements of a given bottleneck Monge matrix such that all elements become distinct and hence the resulting matrix becomes a strict bottleneck Monge matrix. Such a perturbation, does, however, not exist in the general case as can be seen from the example matrix C given above.

4.2. Alternative characterizations of bottleneck Monge matrices

In the sequel we discuss alternative characterizations of bottleneck Monge matrices. We start with bottleneck Monge matrices with only two rows (or two columns). The following lemma from Klinz et al. [87] builds upon ideas of Burkard [31], Johnson [79] and Szwarc [121].

Lemma 4.2. *Let C be a $2 \times n$ matrix and define the following three sets of columns: $J^< := \{j : c_{1j} < c_{2j}\}$, $J^= := \{j : c_{1j} = c_{2j}\}$ and $J^> := \{j : c_{1j} > c_{2j}\}$. Then C is a bottleneck Monge matrix if and only if column j precedes column s , i.e. $j < s$, whenever one of the following three conditions is satisfied:*

- (i) $j \in J^<$ and $s \in J^>$,
- (ii) $j \in J^>$, $s \notin J^<$ and $c_{1j} < c_{1s}$,
- (iii) $j \in J^<$, $s \notin J^>$ and $c_{2j} > c_{2s}$.

An immediate consequence of Lemma 4.2 is that it can be verified in $O(n)$ time whether a given $2 \times n$ matrix is bottleneck Monge. Thus an $O(nm \min\{n, m\})$ time algorithm for recognizing $m \times n$ bottleneck Monge matrices follows: simply check every $2 \times n$ or every $m \times 2$ submatrix. It is an open question whether this result can be improved. (Recall that Monge matrices can be recognized in $O(nm)$ time.)

Let us now turn to general $m \times n$ bottleneck Monge matrices. For a given $m \times n$ matrix C let $\tilde{c}_1 > \tilde{c}_2 > \dots > \tilde{c}_L$ be the sequence of all pairwise distinct entries of C . With each value \tilde{c}_k we associate a 0–1 matrix $T^k = (t_{ij}^k)$, $k = 1, \dots, L$, by setting $t_{ij}^k = 0$ if $c_{ij} \leq \tilde{c}_k$ and $t_{ij}^k = 1$ otherwise. The following characterization of bottleneck Monge matrices in terms of the matrices T^k is due to Klinz et al. [87].

Observation 4.3. *C is a bottleneck Monge matrix if and only if all matrices T^k , $k = 1, \dots, L$, are bottleneck Monge matrices.*

Observation 4.3 motivates to investigate the class of 0–1 bottleneck Monge matrices. In the sequel two characterizations of 0–1 bottleneck Monge matrices from the literature are summarized.

In order to state the first characterization we need a few definitions. An undirected graph $G = (V, E)$ is called a *permutation graph*, if there exists a pair (ρ_1, ρ_2) of permutations of the vertex set V such that $(i, j) \in E$ iff vertex i precedes vertex j in ρ_1 and j precedes i in ρ_2 .

A bipartite graph $G = (V_1, V_2; E)$ is said to be *strongly ordered* if for all $(i, j), (i', j') \in E$, where $i, i' \in V_1$ and $j, j' \in V_2$, it follows from $i < i'$, $j' < j$ that $(i, j') \in E$ and $(i', j) \in E$.

Given a 0–1 matrix C , the *bipartite graph* $B(C) = (V_1, V_2; E_C)$ associated with C is defined as follows: There is a vertex in V_1 for each row of C , a vertex in V_2 for each column of C and an edge $(i, j) \in E_C$ joining vertices $i \in V_1$ and $j \in V_2$ iff $c_{ij} = 1$.

The following observation follows directly from Theorem 3 in Chen and Yesha [45].

Observation 4.4. *A 0–1 matrix C is a bottleneck Monge matrix if and only if its associated bipartite graph $B(C)$ is the complement of a strongly ordered bipartite permutation graph.*

0–1 bottleneck Monge matrices can also be characterized in a more direct way. Let C be an $m \times n$ 0–1 matrix with no rows of all ones. Denote by s_i resp. f_i the position of the first resp. last zero in row i . A 0–1 Matrix C is said to be a *double staircase matrix* if $s_1 \leq s_2 \leq \dots \leq s_n$, $f_1 \leq f_2 \leq \dots \leq f_n$ and $a_{ij} = 0$ for all $j \in [s_i, f_i]$. (Matrices with a similar property were introduced independently in Chen and Yesha [45].) Then we obtain (cf. [87]):

Theorem 4.5. *Let C be a 0–1 matrix with no all ones rows and no all ones columns. Then C is a bottleneck Monge matrix if and only if it is a double staircase matrix.*

Theorem 4.5 implies that a 0–1 bottleneck Monge matrix necessarily must have the consecutive zeros property for both rows and columns (cf. [125]). This is, however, not sufficient, but Klinz et al. [87] show that in order to obtain the class of 0–1 bottleneck Monge matrices only two 3×4 matrices and their transposed matrices have to be added to the set of forbidden submatrices from Tucker's characterization of the class of 0–1 matrices with consecutive zeros property for rows and columns.

To conclude this subsection, let us mention that in analogy to the sum case there is also a continuous characterization of bottleneck Monge matrices. We simply have to replace “+” by “max” in property (14), i.e. let $f : \mathbb{R}^2 \mapsto \mathbb{R}$ be a function satisfying

$$\max \{f(x, y), f(x', y')\} \leq \max \{f(x', y), f(x, y')\} \quad \text{for all } x \leq x', y \leq y'. \quad (33)$$

Then the matrix $C = (c_{ij})$ generated by $c_{ij} = f(i, j)$, $i = 1, \dots, m$, $j = 1, \dots, n$ is bottleneck Monge. The class of functions satisfying (33) has, however, at least to our knowledge not been studied in the literature.

4.3. Applications of bottleneck Monge matrices

4.3.1. Bottleneck assignment and transportation problems

Let a real $m \times n$ cost matrix $C = (c_{ij})$ and a nonnegative supply vector $a = (a_1, \dots, a_m)$ and a nonnegative demand vector $b = (b_1, \dots, b_n)$ such that $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ be given. The *bottleneck transportation problem* (BTP) which is also known as *time transportation problem* is to find a feasible transportation schedule which minimizes the largest shipping cost of a single item. More formally, the (BTP) can be written as

$$\min \max_{x_{ij} > 0} c_{ij} \quad (34)$$

$$\text{s.t. } (17) - (19). \quad (35)$$

Using analogous arguments as in the case of the Hitchcock transportation problem one can prove the following theorem (cf. [31]).

Theorem 4.6. *The north-west corner rule produces an optimal solution of (BTP) for all supply and demand vectors a and b with $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ if and only if the cost matrix C is a bottleneck Monge matrix.*

Thus if the cost matrix C is (inverse) bottleneck Monge the resulting bottleneck transportation problem can be solved in $O(n+m)$ time and its special case, the *bottleneck assignment problem* has again an optimal optimal solution with a fixed structure which does not depend on C . Since Lemma 4.2 implies that the columns of any $2 \times n$ matrix can be reordered such that the resulting matrix becomes a bottleneck Monge matrix, any $2 \times n$ bottleneck transportation problem can be solved in $O(n \log n)$ steps (cf. also [121]). A linear but more involved algorithm for this special case is due to Varadarajan [126].

Obviously, the greedy approach discussed above can be extended to nonbipartite graphs, i.e. to uncapacitated flow problems with bottleneck objective function, in the same way as discussed in Adler et al. [1] for the sum case.

The unbalanced bottleneck transportation problem where $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ can be solved in $O(\log^2 n(m\sqrt{n \log m} + n))$ time if C is an $m \times n$ (inverse) bottleneck Monge matrix with $m \leq n$ and in $O(m \log^2 n)$ time if the rows of C are also bitonic (see [20]).

4.3.2. Bottleneck travelling salesman problems

The *bottleneck travelling salesman problem* (BTSP) is closely related to the classical travelling salesman problem. Given an $n \times n$ cost matrix C , it asks for a tour ϕ which minimizes the largest edge cost $\max\{c_{i\phi(i)} : i = 1, \dots, n\}$.

Burkard and Sandholzer [38] showed that if the cost matrix C of a (BTSP) is bottleneck Monge, there always exists an optimal tour which is pyramidal. Thus an optimal tour can be determined in $O(n^2)$ time by a straightforward modification of the

dynamic programming approach discussed above for the ordinary travelling salesman problem (simply replace in the recurrence formula every occurrence of “+” by “max”).

Since bottleneck Monge matrices are in general not totally monotone, the ideas of Park [103] which lead to a linear time algorithm in the sum case do not work in the bottleneck case. The situation changes, however, if the distance matrix C satisfies the strict bottleneck Monge property (32). In this case the corresponding bottleneck travelling salesman problem can be solved in $O(n)$ time as shown by Bein et al. [20]. (An alternative way to obtain this result is to exploit the relationship between Monge matrices and strict bottleneck Monge matrices.)

The (BTSP) for inverse bottleneck Monge matrices has been mentioned as an open problem in van der Veen [128]. It can, however, be shown that Theorem 3.3 on subtour patching can be extended to the bottleneck case. As a consequence, the bottleneck travelling salesman problem for inverse bottleneck Monge matrices can be solved in $O(nq)$ time where q is the number of subcycles in the optimal bottleneck assignment. For strict bottleneck Monge matrices C the running time can be improved to $O(n)$ by proceeding in the same way as Burkard and Deineko in [34].

4.3.3. Minimum makespan flow-shop scheduling problems

In the flow-shop scheduling problem we are given n jobs, m machines and a fixed ordering π of the jobs according to which the jobs have to pass the machines. Let p_{ij} denote the processing time of job j on machine i . The *minimum makespan flow-shop scheduling problem* is to find a schedule which minimizes the maximum of the completion times of the n jobs.

A famous result of Johnson [79] states that the minimum makespan flow-shop scheduling problem for $m = 2$ machines is solved to optimality by ordering the jobs such that a job j precedes a job s in an optimal schedule if and only if

$$\min \{p_{1j}, p_{2s}\} \leq \min \{p_{1s}, p_{2j}\}.$$

Therefore an optimal schedule in the two-machine case can always be found just by sorting the jobs according to this rule.

Johnson's result cannot be generalized to the case of $m \geq 3$ machines. It leads, however, to a class of efficiently solvable flow-shop scheduling problems. Several authors observed that if the matrix matrix $P = (p_{ij})$ built up from the processing times p_{ij} fulfills the property

$$\min \{p_{ij}, p_{rs}\} \geq \min \{p_{is}, p_{rj}\} \quad \text{for all } 1 \leq i < r \leq m, \quad 1 \leq j < s \leq n \quad (36)$$

then the schedule obtained by processing the jobs on each machine according to the sequence $\langle n, n-1, \dots, 1 \rangle$ is minimal with respect to the makespan criterion (see e.g. the survey paper by Monma and Rinnooy Kan [97]).

Obviously, condition (36) is equivalent to requiring that the matrix $\tilde{P} = (-p_{ij})$ is a bottleneck Monge matrix. Hence Johnson's rule reorders the jobs such that the matrix of the negative processing times becomes bottleneck Monge.

Since the numbering of the machines does not play a role in flow-shop scheduling, the scheduling problem stated above can be solved efficiently not only for inverse bottleneck Monge matrices P , but also for bottleneck Monge matrices P .

4.3.4. Bottleneck path problems

Recall that the algorithm of Pferschy et al. [104] for the longest path problem in undirected bipartite graphs with edge weights satisfying the Monge property was the only one of the path algorithms mentioned in Section 3 which does not exploit total monotonicity. Thus only this algorithm can be extended to the case of weights satisfying the bottleneck Monge property.

The following results only apply to weights with the strict bottleneck Monge property: A path from vertex 1 to vertex n which minimizes the largest edge cost can be found in $O(n)$ time resp. in $O(\min\{kn, n^{3/2} \log^{5/2} n\})$ time if only paths with exactly k edges are feasible. Bein et al. [20] derive these results building on the work of Wilber [133] and Aggarwal et al. [10], respectively.

Similar results can be obtained for edge weights satisfying the inverse strict bottleneck Monge property. We refrain, however, from giving further details.

4.3.5. Further applications

For space restrictions we have not discussed all applications of bottleneck Monge matrices we are aware of. Most of the omitted applications can, however, be obtained directly by replacing the sum objective by a bottleneck objective. If the algorithm for the solution of the corresponding sum problem does not require total monotonicity, then the results for the respective bottleneck problem will be valid for general bottleneck Monge matrices while otherwise they hold only for strict bottleneck Monge matrices.

For example, it is straightforward to see that the max-analogues of the dynamic programming recurrences discussed in Section 3 can be solved in the same time as their sum equivalents as long as the involved weights satisfy the strict bottleneck Monge property. Compared to the sum case, however, not so many applications which lead to recurrences of this type are known.

5. Algebraic Monge matrices

5.1. Definitions and fundamental properties of algebraic Monge matrices

In this section the concepts of Monge matrices and bottleneck Monge matrices introduced in Sections 2 and 4 will be embedded into a larger framework.

To this end, let (H, \oplus, \preceq) be a totally ordered commutative semigroup such that \oplus is compatible with \preceq , i.e.

$$a \preceq b \implies a \oplus c \preceq b \oplus c \quad \text{for all } a, b, c \in H. \quad (37)$$

An $m \times n$ matrix C with elements c_{ij} drawn from H is called *algebraic Monge matrix* or *Monge matrix* w.r.t. (H, \oplus, \preceq) if it satisfies the following *algebraic Monge property*:

$$c_{ij} \oplus c_{rs} \preceq c_{is} \oplus c_{rj} \quad \text{for all } 1 \leq i < r \leq m, 1 \leq j < s \leq n. \quad (38)$$

By specializing H , \oplus and \preceq we obtain various special classes of algebraic Monge matrices. Below we give some examples.

Example. (a) *Monge matrices*: Let $H := \mathbb{R}$ endowed with the natural order \leq and $a \oplus b := a + b$. *Monge matrices with infinite entries* result for $H := \overline{\mathbb{R}}$ where $+$ and \leq are extended to $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ in the natural way.

(b) *Bottleneck Monge matrices*: Set $H := \mathbb{R}$, $a \oplus b := \max\{a, b\}$ and take again \leq as order \preceq .

(c) *Inverse (bottleneck) Monge matrices*: H and \oplus are defined as in (a) and (b), respectively, but instead of \leq the reverse order \geq is used.

(d) *Strict bottleneck Monge matrices*: For the ease of exposition we first describe a setting which is slightly more general than required for modelling the strict bottleneck case.

Let H be the set of ordered tuples $(\alpha_1, \dots, \alpha_p)$, $p \geq 1$, with $\alpha_i \in \mathbb{R}$, $1 \leq i \leq p$ and $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_p$. The composition \oplus is defined such that $(\alpha_1, \dots, \alpha_p) \oplus (\beta_1, \dots, \beta_q) = (\gamma_1, \dots, \gamma_{p+q})$, where $\langle \gamma_1, \gamma_2, \dots, \gamma_{p+q} \rangle$ is the sorted sequence obtained by merging the sequences $\langle \alpha_1, \dots, \alpha_p \rangle$ and $\langle \beta_1, \dots, \beta_q \rangle$. The order \preceq is chosen to be the usual lexicographic order, i.e. we have $(\alpha_1, \dots, \alpha_p) \prec (\beta_1, \dots, \beta_q)$ if and only if either $p < q$ and $\alpha_j = \beta_j$ for all $j = 1, \dots, p$ or there exists an $i \in \{1, \dots, p\}$ such that $\alpha_i < \beta_i$ and $\alpha_j = \beta_j$ for all $j = 1, \dots, i-1$.

In the case of strict bottleneck Monge matrices as introduced in [20], H consists only of tuples of length 1 or 2 and all matrix elements are tuples of length 1, i.e. real numbers. Note that this implies that the composition of two matrix elements a and b is given by $a \oplus b := (\max\{a, b\}, \min\{a, b\})$.

(e) *Lexicographic (bottleneck) Monge matrices*: Let H be the set of real vectors with $d \geq 2$ components endowed with the lexicographic order. Depending on whether \oplus is taken to be the usual vector addition or the componentwise maximum operation we obtain lexicographic Monge matrices and lexicographic bottleneck Monge matrices, respectively.

An important subclass of algebraic Monge matrices results if the composition \oplus satisfies the following so-called *strong cancellation rule*, i.e.

$$a \oplus c = b \oplus c \implies a = b \quad \text{for all } a, b, c \in H. \quad (39)$$

Examples for such matrices are, e.g. Monge matrices and strict bottleneck Monge matrices, while (39) is violated for Monge matrices with infinite entries as well as for bottleneck Monge matrices.

In conjunction with the compatibility axiom (37) the strong cancellation rule implies

$$a \oplus c \preceq b \oplus c \implies a \preceq b \quad \text{for all } a, b, c \in H. \quad (40)$$

Hence requiring (39) is equivalent to requiring the property

$$a \prec b \implies a \oplus c \prec b \oplus c \quad \text{for all } a, b, c \in H, \quad (41)$$

where $a \prec b$ denotes the situation that $a \preceq b$, but $a \neq b$. Property (41) is referred to as *strict compatibility* in Bein et al. [20].

It turns out that algebraic Monge matrices in strongly cancellative semigroups (i.e. semigroups (H, \oplus) for which the strong cancellation rule holds) have many nice properties in common with Monge matrices. Many of these properties do not hold for general algebraic Monge matrices. The lemma below is an immediate consequence of property (40).

Lemma 5.1. *Let C be a matrix over (H, \oplus, \preceq) such that (39) holds. Then we have*

(i) *C is an algebraic Monge matrix if and only if*

$$c_{ij} \oplus c_{i+1,j+1} \preceq c_{i+1,j} \oplus c_{i,j+1} \quad \text{for all } 1 \leq i < m, 1 \leq j < n. \quad (42)$$

(ii) *If C is an algebraic Monge matrix, then C is totally monotone, i.e.*

$$c_{is} \prec c_{ij} \implies c_{rs} \prec c_{rj} \quad \text{for all } 1 \leq i < r \leq m, 1 \leq j < s \leq n.$$

The strong cancellation rule is necessary for Lemma 5.1 to hold as can be seen by the example of bottleneck Monge matrices which fulfill neither of the two claims (i) and (ii) above.

Monge matrices with infinite entries and bottleneck Monge matrices are, however, examples for classes of algebraic Monge matrices which fulfill a weakened version of (39), the so-called *weak cancellation rule*.

$$a \oplus c = b \oplus c \implies a = b \text{ or } a \oplus c = c \quad \text{for all } a, b, c \in H. \quad (43)$$

5.2. Applications of algebraic Monge matrices

5.2.1. Algebraic transportation problems

Both the classical Hitchcock transportation problem and the bottleneck transportation problem fit into the much larger class of algebraic transportation problems introduced in Burkard [28]. The *algebraic transportation problem* (ATP) can be formulated as follows:

Let $(R, +)$ be a subsemigroup of $(\mathbb{R}_0^+, +)$ and denote by P_R the set of all transportation schedules (x_{ij}) with $x_{ij} \in R$, $i = 1, \dots, m$, $j = 1, \dots, n$, which fulfill the usual transportation constraints (17)–(19). In order to define the cost of a transportation schedule, we need to introduce an outer composition $\otimes : H \times R \mapsto H$ which satisfies the following distributive laws

$$(a \oplus b) \otimes z = (a \otimes z) \oplus (b \otimes z) \quad \text{for all } a, b \in H \text{ and } z \in R \quad (44)$$

$$a \otimes (z + z') = (a \otimes z) \oplus (a \otimes z') \quad \text{for all } a \in H \text{ and } z, z' \in R \quad (45)$$

and is compatible with \preceq , i.e.

$$a \preceq b \implies a \otimes z \preceq b \otimes z \quad \text{for all } a, b \in H \text{ and } z \in R. \quad (46)$$

Given a cost matrix $C = (c_{ij})$ with entries c_{ij} drawn from H , the (ATP) is to minimize the objective function

$$(c_{11} \otimes x_{11}) \oplus (c_{12} \otimes x_{12}) \oplus \dots \oplus (c_{mn} \otimes x_{mn}) \quad (47)$$

over all feasible solutions $X = (x_{ij}) \in P_R$ (note that the minimization is w.r.t. to the order \preceq).

Example. In all examples below we take $R := \mathbb{R}_0^+$, but other choices are possible as well, e.g. setting $R := \mathbb{N}_0$ leads to integral transportation problems.

(a) *Hitchcock transportation problem:* Let $(H, \oplus, \preceq) := (\overline{\mathbb{R}}, +, \leq)$ and let \otimes be the multiplication of reals extended to $\overline{\mathbb{R}}$ by setting $\infty \cdot 0 = 0$ and $\infty \cdot z = \infty$ for $z > 0$.

(b) *Bottleneck transportation problem:* Let $(H, \oplus, \preceq) := (\mathbb{R} \cup \{-\infty\}, \max, \leq)$ and define $a \otimes z := a$ for $z > 0$ and $a \otimes 0 := -\infty$.

(c) *Time-cost transportation problem:* Given two cost matrices C and C' , determine a feasible transportation schedule (x_{ij}) which minimizes $\sum_{i=1}^m \sum_{j=1}^n c'_{ij} \delta(x_{ij})$ among all solutions which minimize $z(x) = \max_{x_{ij} > 0} c_{ij}$, where $\delta(x_{ij}) = 0$ if $c_{ij} < z(x)$ and $\delta_{ij}(x_{ij}) = 1$ otherwise.

Let $H := (\mathbb{R} \cup \{-\infty\}) \times \mathbb{R}$ and let \preceq be the lexicographic order. \oplus and \otimes are defined as follows:

$$(a, b) \oplus (a', b') := \begin{cases} (a, b) & \text{if } a > a', \\ (a', b') & \text{if } a < a', \\ (a, b + b') & \text{if } a = a', \end{cases} \quad (a, b) \otimes z := \begin{cases} (-\infty, 0) & \text{if } z = 0, \\ (a, bz) & \text{if } z > 0. \end{cases}$$

For further examples and a detailed discussion of the (ATP) we refer to Burkard [28] and Burkard and Zimmermann [40] where a general solution approach is given (cf. also [138] for more general algebraic optimization problems). In the following we will deal only with the special case of the (ATP) where the cost matrix C is algebraic Monge.

The following theorem is a generalization of Theorem 3.1 on the Hitchcock transportation problem and of Theorem 4.6 on the bottleneck transportation problem.

Theorem 5.2. *The north-west corner rule yields an optimal solution of the algebraic transportation problem with cost matrix C for all feasible supply and demand vectors if and only if C is an algebraic Monge matrix.*

Proof. “ \Rightarrow ”: Henceforth we adopt the convention to regard a feasible solution (x_{ij}) of the (ATP) as a vector in R^{mn} where the components x_{ij} are arranged rowwise in the order $(x_{11}, x_{12}, \dots, x_{1n}, x_{21}, \dots, x_{2n}, \dots, x_{m1}, \dots, x_{mn})$.

Let us assume that the north-west corner rule produces the solution vector (x_{ij}) and let (y_{ij}) be the uniquely defined optimal solution of (ATP) for which the vector

(δ_{ij}) defined by $\delta_{ij} = x_{ij} - y_{ij}$ is lexicographically minimal. We have to show that all components δ_{ij} are zero.

Assume the contrary and let δ_{pq} be the first nonzero component of the vector (δ_{ij}) . Since (x_{ij}) is obtained from the north-west corner rule, we have $x_{pq} > y_{pq}$ and due to the transportation constraints there exists an index $r > p$ with $y_{rq} > 0$ and an index $s > q$ with $y_{ps} > 0$. Let $\Delta := \min\{x_{pq} - y_{pq}, y_{rq}, y_{ps}\} > 0$ and define a new feasible solution (\hat{y}_{ij}) such that

$$\hat{y}_{ij} := \begin{cases} y_{ij} - \Delta & \text{if } (i,j) = (r,q) \text{ or } (i,j) = (p,s), \\ y_{ij} + \Delta & \text{if } (i,j) = (p,q) \text{ or } (i,j) = (r,s), \\ y_{ij} & \text{otherwise.} \end{cases}$$

Let $c(y)$ denote the objective function value corresponding to the solution (y_{ij}) . If we can show that $c(\hat{y}) \preceq c(y)$, then we are finished, since then the vector $(\hat{\delta}_{ij})$ with $\hat{\delta}_{ij} = x_{ij} - \hat{y}_{ij}$ is lexicographically smaller than (δ_{ij}) which contradicts the choice of (y_{ij}) . Hence it only remains to be shown that $c(\hat{y}) \preceq c(y)$.

To this end, we proceed as follows: By the associativity of \oplus and the compatibility assumption (37) it suffices to show that

$$(c_{pq} \otimes (y_{pq} + \Delta)) \oplus (c_{rs} \otimes (y_{rs} + \Delta)) \oplus (c_{ps} \otimes (y_{ps} - \Delta)) \oplus (c_{rq} \otimes (y_{rq} - \Delta)) \preceq (c_{pq} \otimes y_{pq}) \oplus (c_{rs} \otimes y_{rs}) \oplus (c_{ps} \otimes y_{ps}) \oplus (c_{rq} \otimes y_{rq}) \quad (48)$$

since all the other terms in $c(y)$ and $c(\hat{y})$ are identical. Attaching the term $(c_{ps} \otimes \Delta) \oplus (c_{rq} \otimes \Delta)$ to the right of both sides of (48) yields

$$(c_{pq} \otimes (y_{pq} + \Delta)) \oplus (c_{rs} \otimes (y_{rs} + \Delta)) \oplus (c_{ps} \otimes (y_{ps} - \Delta)) \oplus (c_{rq} \otimes (y_{rq} - \Delta)) \oplus (c_{ps} \otimes \Delta) \oplus (c_{rq} \otimes \Delta), \quad (49)$$

$$= (c_{pq} \otimes y_{pq}) \oplus (c_{pq} \otimes \Delta) \oplus (c_{rs} \otimes y_{rs}) \oplus (c_{rs} \otimes \Delta) \oplus (c_{ps} \otimes y_{ps}) \oplus (c_{rq} \otimes y_{rq}) \oplus (c_{ps} \otimes \Delta) \oplus (c_{rq} \otimes \Delta), \quad (50)$$

$$\preceq (c_{pq} \otimes y_{pq}) \oplus (c_{rs} \otimes y_{rs}) \oplus (c_{ps} \otimes y_{ps}) \oplus (c_{rq} \otimes y_{rq}) \oplus (c_{ps} \otimes \Delta) \oplus (c_{rq} \otimes \Delta),$$

where the step from (49) to (50) follows from applying the second distributive law (45). Thanks to axiom (37) and the first distributive law (44) the above inequality holds only if

$$(c_{pq} \otimes \Delta) \oplus (c_{rs} \otimes \Delta) = (c_{pq} \oplus c_{rs}) \otimes \Delta \preceq (c_{ps} \otimes \Delta) \oplus (c_{rq} \otimes \Delta) = (c_{ps} \oplus c_{rq}) \otimes \Delta. \quad (51)$$

Since C is an algebraic Monge matrix we have $c_{pq} \oplus c_{rs} \preceq c_{ps} \oplus c_{rq}$ which implies that inequality (51) holds. Combining all steps from above we arrive at the conclusion that $c(\hat{y}) \leq c(y)$ and thus we are done.

“ \Leftarrow ”: Suppose that there exist rows p and r , $p < r$ and columns q and s , $q < s$, such that $c_{ps} \oplus c_{rq} \prec c_{pq} \oplus c_{rs}$. Setting a_p, a_r, b_q and b_s to one and all other supplies and demands to zero, yields an instance of (ATP) for which the north-west corner rule yields the solution $x_{pq} = 1$ and $x_{rs} = 1$ with cost $c_{pq} \oplus c_{rs}$ while the optimum solution is $x_{ps} = 1$ and $x_{rq} = 1$ with cost $c_{ps} \oplus c_{rq}$. \square

We note that the theorem above and the essential parts of its proof are implicit in the work of Burkard [27]. The proof given here is more direct and fits better into the framework of this paper.

Theorem 5.2 implies that if the cost matrix C of an (ATP) is an (inverse) algebraic Monge matrix, an optimal solution can be found in $O(n+m)$ time. If also the optimal objective function value is required, then additionally $(m+n-1)$ operations of type \oplus need to be performed.

Algebraic transportation problems with Monge cost matrices are not only of theoretical interest, they arise, e.g. in the computation of lower bounds for certain classes of scheduling problems with algebraic objective functions (cf. [27]).

5.2.2. Algebraic travelling salesman problems

Given a cost matrix C whose entries are drawn from H , the *algebraic travelling salesman problem* (ATSP) is to find a tour ϕ which minimizes $\oplus_{1 \leq i \leq n} c_{i\phi(i)}$. Obviously, the classical sum travelling salesman problem and the bottleneck travelling salesman problem are special cases of (ATSP). For further examples the reader is referred to Burkard and van der Veen [39]. In this paper Burkard and van der Veen show that if C is an algebraic Monge matrix, then the corresponding (ATSP) always has a pyramidal optimal tour. Hence the (ATSP) restricted to algebraic Monge matrices can be solved in $O(n^2)$ steps where each \oplus operation and each comparison w.r.t. \preceq counts as one step. If the algebraic system (H, \oplus, \preceq) satisfies the strong cancellation rule (39), which implies the total monotonicity of the cost matrix C , Park's idea from [103] leads to a reduction of the number of required steps to $O(n)$.

The (ATSP) for inverse algebraic Monge matrices can be handled in the same way as the bottleneck travelling salesman problem. Again the subtour patching approach originally developed for the sum case can be applied and yields an algorithm which takes $O(n^2)$ steps for general inverse algebraic Monge matrices and $O(n)$ steps for inverse algebraic Monge matrices in strongly cancellative semigroups.

5.2.3. Algebraic path problems

Given weights w_{ij} drawn from (H, \oplus, \preceq) , the *algebraic path problem* asks for a path P such that $\oplus_{(i,j) \in P} c_{ij}$ is minimized. It is straightforward to show that the results on path problems with sum and bottleneck objective function mentioned in Sections 3 and 4 can be extended to the algebraic setting. In those cases where total monotonicity is required, the results hold again only for algebraic Monge matrices in strongly cancellative semigroups. For example, a shortest 1 to n path in a graph with edge weights

satisfying the algebraic Monge property in a strongly cancellative semigroup can be found in $O(n)$ steps (see [20]).

6. Monge arrays

6.1. Definitions

The concept of Monge matrices has been generalized to higher dimensions independently by several authors in different areas of mathematics resp. computer science.

We start with a definition due to Aggarwal and Park [5, 6]. An $n_1 \times n_2 \times \cdots \times n_d$ array with real entries $c[i_1, i_2, \dots, i_d]$, $i_k = 1, \dots, n_k$, $k = 1, \dots, d$, is called a (d -dimensional) *Monge array* if for all $i_k = 1, \dots, n_k$ and $j_k = 1, \dots, n_k$, $k = 1, \dots, d$, we have

$$c[s_1, s_2, \dots, s_d] + c[t_1, t_2, \dots, t_d] \leq c[i_1, i_2, \dots, i_d] + c[j_1, j_2, \dots, j_d], \quad (52)$$

where $s_k := \min\{i_k, j_k\}$ and $t_k := \max\{i_k, j_k\}$, $k = 1, \dots, d$. Obviously this definition can be extended to cover infinite entries in the same way as in the two-dimensional case.

Inverse Monge arrays result by reversing the inequality sign in (52). An inverse Monge array C can be transformed into a Monge array C' by multiplying each of its entries by -1 , but unfortunately, unlike in the two-dimensional case, it is in general not possible to find permutations π_k , $k = 1, \dots, d$ such that the permuted array C_{π_1, \dots, π_d} with entries $c[\pi_1(i_1), \dots, \pi_d(i_d)]$ becomes a Monge array.

Note that for $d = 2$ property (52) yields the class of (inverse) Monge matrices. The case $d = 3$ can also be written in the following equivalent form:

$$c[i_1, j_1, k_1] + c[i_2, j_2, k_2] \leq \begin{cases} c[i_1, j_1, k_2] + c[i_2, j_2, k_1] \\ c[i_1, j_2, k_1] + c[i_2, j_1, k_2] \\ c[i_2, j_1, k_1] + c[i_1, j_2, k_2] \end{cases} \quad (53)$$

for all $1 \leq i_1 \leq i_2 \leq n_1$, $1 \leq j_1 \leq j_2 \leq n_2$ and $1 \leq k_1 \leq k_2 \leq n_3$.

Similarly as in the case $d = 2$ a subclass of Monge arrays can be generated by borrowing an idea from statistics. Let D be an $n_1 \times \cdots \times n_d$ nonnegative array with entries $d[i_1, i_2, \dots, i_d]$. Then the array C obtained by

$$c[i_1, i_2, \dots, i_d] = - \sum_{\ell_1=1}^{i_1} \sum_{\ell_2=1}^{i_2} \cdots \sum_{\ell_d=1}^{i_d} d[\ell_1, \ell_2, \dots, \ell_d] \quad \text{for all } 1 \leq i_k \leq n_k, k = 1, \dots, d \quad (54)$$

is called the *distribution array* induced by the *density array* D . It is easy to check that C is a Monge array and that $-C$ is an inverse Monge array (see also [21]).

Unlike in the two-dimensional case, for $d \geq 3$ we cannot get rid of the minus sign in (54) by changing the range of the summation indices ℓ_1, \dots, ℓ_d . It still holds, however, that, given a distribution array or Monge array D and vectors $U_1 = (u_1[i_1]), \dots, U_d =$

$(u_d[i_d])$, the array C with entries $c[i_1, \dots, i_d] := d[i_1, \dots, i_d] + u_1[i_1] + \dots + u_d[i_d]$ is a Monge array. The converse, however, is not any longer true in general for $d \geq 3$. As an example of a Monge array C for which there does not exist a distribution array D and vectors U_1, \dots, U_d such that $c[i_1, \dots, i_d] := d[i_1, \dots, i_d] + u_1[i_1] + \dots + u_d[i_d]$ consider the $2 \times 2 \times 2$ array C with lower plane $C_1 = (c[i_1, i_2, 1])$ and upper plane $C_2 = (c[i_1, i_2, 2])$ given by

$$C_1 = \begin{pmatrix} 0 & 3 \\ 1 & 4 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 1 & 3 \\ 1 & 0 \end{pmatrix}.$$

For conditions under which a Monge array is also a distribution array, the reader is referred to Bein et al. [21].

6.2. Alternative characterizations of Monge arrays

As in the case of Monge matrices, Monge arrays and submodular functions are intimately related (see also [105]).

Observation 6.1. Let $\mathcal{I} = \{1, \dots, n_1\} \times \{1, \dots, n_2\} \times \dots \times \{1, \dots, n_d\}$. C is a Monge array (resp. an inverse Monge array) if and only if the function $f : \mathcal{I} \mapsto \mathbb{R}$ with $f(i_1, \dots, i_d) := c[i_1, \dots, i_d]$ is submodular (resp. supermodular) on the lattice $\mathcal{L} = (\mathcal{I}, \vee, \wedge)$ where \vee and \wedge are defined as in Section 2, i.e. $(i_1, i_2, \dots, i_d) \vee (j_1, j_2, \dots, j_d) := (\max\{i_1, j_1\}, \max\{i_2, j_2\}, \dots, \max\{i_d, j_d\})$ and $(i_1, i_2, \dots, i_d) \wedge (j_1, j_2, \dots, j_d) := (\min\{i_1, j_1\}, \min\{i_2, j_2\}, \dots, \min\{i_d, j_d\})$.

Another class of submodular functions which play an important role in combinatorial optimization are submodular set-functions. A function $f : 2^S \mapsto \mathbb{R}$ defined on the subsets of a finite set S is said to be a *submodular set-function* if $f(S' \cap S'') + f(S' \cup S'') \leq f(S') + f(S'')$ holds for any two subsets S' and S'' of S .

Since, as shown by Lovász [92], submodular set-functions are closely connected to convex functions, the following observation of Park [102] relates Monge arrays, submodular set-functions and convexity.

Observation 6.2. Let C be a d -dimensional $2 \times 2 \times \dots \times 2$ array and let $f_C : 2^S \mapsto \mathbb{R}$ with $S = \{1, \dots, d\}$ denote the set-function which is associated with C by setting $f_C(S') := c[i_1, i_2, \dots, i_d]$, where $i_k = 1$ if $k \notin S'$ and $i_k = 2$ if $k \in S'$ for all $1 \leq k \leq d$. Then C is a d -dimensional Monge array if and only if f_C is a submodular set-function.

A third characterization of Monge arrays can be given in terms of its 2-dimensional subarrays (cf. [6]).

Lemma 6.3. A d -dimensional real array C is an (inverse) Monge array if and only if every two-dimensional submatrix of C is an (inverse) Monge matrix.

As noted by Queyranne et al. [105], this lemma follows immediately from a general result on submodular functions in Topkis [124]. Lemma 6.3 becomes wrong when C is allowed to contain infinite entries. In that case requiring the Monge property for every two-dimensional submatrix is not sufficient to ensure that the d -dimensional Monge property holds.

As in the two-dimensional case checking whether a given array C is a Monge array becomes often easier if C is generated by setting $c[i_1, \dots, i_d] := f(i_1, \dots, i_d)$ where f is a function from \mathbb{R}^d onto the reals. Obviously, f is submodular (supermodular) w.r.t. $(\mathbb{R}^d, \vee, \wedge)$ if and only if f satisfies property (14) (resp. (15) introduced in Section 2 with respect to any two of its arguments. Thus, if the mixed second partial derivatives of f exist, f is submodular resp. supermodular on \mathbb{R}^d if and only if $\partial^2 f / \partial x_i \partial x_j$ is ≤ 0 resp. ≥ 0 for all arguments $x \in \mathbb{R}^d$ and all pairs $i, j = 1, \dots, d$, $i \neq j$.

The class of functions which are submodular resp. supermodular on the lattice $(\mathbb{R}^d, \vee, \wedge)$ plays an important role in various branches of mathematics, in particular in probability theory (cf. also Section 9). Below we list several members of this interesting class of functions: (1) $f(x_1, \dots, x_d) := x_1 + \dots + x_d$, (2) $f(x_1, \dots, x_d) := (x_1 + \dots + x_d)^2$, (3) $f(x_1, \dots, x_d) := \min\{x_1, \dots, x_d\}$, (4) $f(x_1, \dots, x_d) := \max\{x_1, \dots, x_d\}$, (5) $f(x_1, \dots, x_d) := x_1 \cdot x_2 \cdot \dots \cdot x_d$, (6) any distribution function of a nonpositive or nonnegative measure and (7) $f(x_1, \dots, x_d) := \tilde{f}(h_1(x_1), \dots, h_d(x_d))$, where \tilde{f} is submodular (supermodular) and h_1, \dots, h_d are real functions which are monotone in the same direction.

6.3. Fundamental properties of Monge arrays

In this subsection we summarize several important properties of Monge arrays which play a fundamental role in applications.

It is easy to see that properties (ii)–(iv) in Observation 2.2 can be generalized to d -dimensional (inverse) Monge arrays. In other words, the class of $n_1 \times \dots \times n_d$ (inverse) Monge arrays is closed under additions and multiplications with a positive scalar.

Following Aggarwal and Park [6] we generalize in the following the notions of monotone resp. totally monotone matrices to d -dimensional arrays. Let C be an $n_1 \times \dots \times n_d$ array. For fixed i_1 , $1 \leq i_1 \leq n_1$, let $j_2(i_1), \dots, j_d(i_1)$ be the second through d th coordinate of the first minimum entry in the $(d-1)$ -dimensional subarray of C which results when fixing the first coordinate to i_1 , i.e.

$$c[i_1, j_2(i_1), j_3(i_1), \dots, j_d(i_1)] = \min_{\substack{1 \leq i_k \leq n_k \\ k=2,\dots,d}} c[i_1, i_2, \dots, i_d],$$

where in case of ties $(j_2(i_1), \dots, j_d(i_1))$ is chosen lexicographically minimal. Analogously, let the indices $j'_2(i_1), \dots, j'_d(i_1)$ be such that

$$c[i_1, j'_2(i_1), j'_3(i_1), \dots, j'_d(i_1)] = \max_{\substack{1 \leq i_k \leq n_k \\ k=2,\dots,d}} c[i_1, i_2, \dots, i_d]$$

and $(j'_2(i_1), \dots, j'_d(i_1))$ is lexicographically minimum. Henceforth the entries $c[i_1, j_2(i_1), \dots, j_d(i_1)]$ and $c[i_1, j'_2(i_1), \dots, j'_d(i_1)]$, respectively, will be referred to as *plane minima* resp. *plane maxima*.

An $n_1 \times \dots \times n_d$ array C is said to be *monotone* if we have either

$$j_k(1) \leq j_k(2) \leq \dots \leq j_k(n_1) \quad \text{for all } k = 2, \dots, n \quad (55)$$

or

$$j'_k(1) \leq j'_k(2) \leq \dots \leq j'_k(n_1) \quad \text{for all } k = 2, \dots, n \quad (56)$$

and if every $(d-1)$ -dimensional subarray of C (which results when fixing one of the d coordinates) is monotone, too. The array C is *totally monotone* if every d -dimensional $2 \times 2 \times \dots \times 2$ subarray of C is monotone.

Lemma 6.4. (cf. [6]) *Monge arrays and inverse Monge arrays are totally monotone arrays. In particular, Monge arrays fulfill (55) and inverse Monge arrays fulfill (56).*

Similar properties as (55) and (56) hold for (inverse) Monge arrays if any fixed number of indices is fixed and we look for the first minimum entry resp. the first maximum entry in the subarray corresponding to the remaining indices (for further details we refer to [102]).

6.4. Applications of Monge arrays

6.4.1. Multidimensional assignment and transportation problems

Since Monge matrices play an important role in greedily solvable assignment and transportation problems, it is not surprising that the same is true for Monge arrays and multidimensional transportation and assignment problems.

In the literature several generalizations of the classical Hitchcock transportation problem to higher dimensions have been studied. In the following we consider the so-called *axial d-dimensional transportation problem* (dTTP) which can be formulated as follows:

$$\begin{aligned} & \min \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_d=1}^{n_d} c[i_1, i_2, \dots, i_d] x_{i_1 i_2 \dots i_d} \\ & \text{s.t. } \sum_{\substack{i_1, i_2, \dots, i_d \\ i_k=q}} x_{i_1 i_2 \dots i_d} = a_q^k \quad \text{for all } k = 1, \dots, d \text{ and } q = 1, \dots, n_k, \\ & \quad x_{i_1 i_2 \dots i_d} \geq 0 \quad \text{for all } i_1, i_2, \dots, i_d. \end{aligned}$$

Therein C is a given $n_1 \times \dots \times n_d$ cost array and a^1, \dots, a^d are given nonnegative supply (demand) vectors with $a^k \in \mathbb{R}^{n_k}$, $k = 1, \dots, d$, such that $\sum_{q=1}^{n_k} a_q^k = \alpha$ for all $k = 1, \dots, d$ with $\alpha > 0$.

Clearly, for any fixed d the linear program (dTP) can be solved in polynomial time. If the cost array is Monge, however, a simple greedy approach determines an optimal solution of (dTP) in $O(d \sum_{i=1}^d n_i)$ time. This greedy approach which will henceforth be referred to as “lexicographical greedy algorithm” is a natural extension of the north-west corner rule and investigates the variables x_{i_1, \dots, i_d} according to the lexicographic order and maximizes each variable in turn.

The theorem given below is a natural generalization of Hoffman’s classical result on the Hitchcock transportation problem (cf. Theorem 3.1). It seems to be first noted by Balinski and Rachev [17], who established the connection to a more general result in probability theory (cf. Section 9, Rachev [107] and Tchen [122]). Later Bein et al. [21] rediscovered the theorem in a purely discrete setting.

Theorem 6.5. *The lexicographical greedy algorithm determines an optimal solution of problem (dTP) for all feasible right-hand side vectors a^k , $k = 1, \dots, d$ if and only if the underlying cost array C is a Monge array.*

We now turn to the *axial d-dimensional assignment problem* (dAP) which is closely related to the (dTP). In the (dAP) we have $a_i^k = 1$ for all $k = 1, \dots, d$ and $i = 1, \dots, n_k$, but additionally we require that all variables x_{i_1, \dots, i_d} are integral. (Note that this requirement is automatically fulfilled in the two-dimensional case.) It is well known that the (dAP) is NP-hard in general, but if the cost array C is a Monge array, then the identity permutation is an optimal solution of the (dAP) (this follows immediately from Theorem 6.5).

As already mentioned in Section 2, the same greedy approach which solves the problems (TP) and (AP) for Monge matrices works also for inverse Monge matrices (because any inverse Monge matrix can be transformed into a Monge matrix by reversing the order of its columns). Unfortunately, this is not any longer true for $d \geq 3$, and even worse, Burkard et al. [37] have shown that the (dAP) remains NP-hard if the cost array C is inverse Monge. Their proof uses cost coefficients $c[i_1, i_2, i_3]$ of the form $\alpha_{i_1}\beta_{i_2}\gamma_{i_3}$ which clearly leads to a special inverse Monge array (the sequences formed by the numbers α_i , β_j resp. γ_k can w.l.o.g. assumed to be nondecreasing). An immediate consequence of this result is that the maximization version of the (dAP), $d \geq 3$, remains NP-hard for Monge arrays C .

6.4.2. Further classes of greedily solvable linear programs

Queyranne et al. [105] considered a dual pair of linear programs where the primal variables are associated to the elements of a sublattice \mathcal{L}' of a finite product lattice \mathcal{L} . They showed that a straightforward greedy approach solves the primal and the dual problem if the cost coefficients associated with the primal variables are submodular. This result has a wide range of applicability, e.g. the multidimensional axial transportation problem including the case of cost arrays with infinite entries (“forbidden entries”) and various optimization problems on polymatroids and related submodular polyhedra can be treated within the framework of [105].

In a recent paper by Faigle and Kern [59] which deals with submodular linear programs on rooted forests an even more general model is proposed. The results of [59] contain the results of [105] as a special case.

6.4.3. Searching in Monge arrays

Similarly as in the two-dimensional case, various searching and selection problems can be solved faster if the underlying arrays are totally monotone. Generalizing the column minimization problem in matrices we obtain the *plane minimization problem* in d -dimensional arrays which is to find the plane minima for all $(d-1)$ -dimensional subarrays (planes) which correspond to a fixed first index i_1 . Given an $n_1 \times \dots \times n_d$ Monge array C all plane minima can be found in $O((\sum_{k=1}^d n_k) \prod_{k=1}^{d-2} \log n_k)$ time (see [5, 6]). Thus the overall minimum entry of C can also be found in $O((\sum_{k=1}^d n_k) \prod_{k=1}^{d-2} \log n_k)$ time.

Another generalization of the column minimization problem is the *tube minimization problem* which is to find the (lexicographically first) minimum in each $(d-2)$ -dimensional subarray resulting when the first two indices i_1 and i_2 are fixed. This problem can be reduced to the plane minimization problem and solved in the same time.

Obviously, the same algorithms work for determining the *plane* and *tube maxima* in inverse Monge arrays. Contrary to the case $d = 2$, the results above, however, do not carry over to computing the plane and tube maxima in Monge arrays resp. the plane and tube minima in inverse Monge arrays. In fact, these problems are harder as shown by Aggarwal and Park [6], who obtained a lower bound of $\Omega(\prod_{k=1}^d n_k / \sum_{k=1}^d (n_k - 1))$ and gave an $O((n_1 + n_2) \prod_{k=3}^d n_k)$ time algorithm.

Further details and related material on searching problems and parallel algorithms for finding plane and tube minima can be found in Aggarwal and Park [5–7] and Park [103].

6.4.4. Applications of searching in Monge arrays

Many geometric problems lead to searching problems in Monge arrays, see Park [102] and Aggarwal and Park [5–7]. For example, the problem of computing for a given convex n -gon an inscribed d -gon with maximum perimeter which has been solved by applying the SMAWK-algorithm in Aggarwal et al. [4], can also be solved by finding the maximum entry in a specially structured d -dimensional Monge array (cf. Aggarwal and Park [6]). Using similar techniques Aggarwal and Park [6] obtained an $O(nd + n \log n)$ time algorithm for computing the minimum-area circumscribing d -gon of a given convex n -gon for the general case where $3 \leq d \leq d$ (the special case $d = 3$ can be solved in $O(n)$ time due to O'Rourke et al. [101]) and an $O(n \log n)$ time algorithm for computing the minimum-perimeter circumscribing triangle.

Another class of problems where searching in Monge arrays turns out to be very helpful are shortest path problems on special grid graphs. Let G be a directed grid graph with vertex set $V := \{v_{ij} : 1 \leq i \leq n_1, 1 \leq j \leq n_2\}$ and edge set E consisting of

all edges of the form $(v_{ij}, v_{i+1,j})$, $(v_{ij}, v_{i,j+1})$ and $(v_{ij}, v_{i+1,j+1})$. The nodes v_{ij} with $i = 1$ or $j = 1$ are called sources and those with $i = n_1$ or $j = n_2$ are called sinks. Given weights on the edges in E , we want to find the shortest path between each pair of sources and sinks. This problem has applications in optimal surface reconstruction from planar contours and in string editing problems in molecular biology (see Aggarwal and Park [7] and the references given therein). The best parallel algorithms known for solving the above shortest path problem on grid graphs are based on a divide-and-conquer approach where in each step the current grid is divided into four sub-grids and the resulting subproblems are then solved recursively. In merging the solutions of the subproblems in an efficient way, finding the tube minima of three-dimensional Monge arrays plays an important role (see [16, 7]).

6.5. Algebraic Monge arrays

The concept of Monge arrays can be extended to an algebraic setting in the same way as done in Section 5 for Monge matrices. As was the case already for $d=2$, some of the properties of Monge arrays do not carry over to general algebraic Monge arrays. For example, a characterization of d -dimensional algebraic Monge arrays in terms of all two-dimensional subarrays is only possible if the underlying algebraic system (H, \oplus, \preceq) satisfies the strong cancellation rule (39). The same assumption is needed to ensure total monotonicity.

Not surprisingly Theorem 6.5 for the d -dimensional axial transportation problem (dTP) can be generalized to the algebraic version of the problem which is defined analogously as the algebraic transportation problem (ATP) in Section 5. Special cases of the algebraic multidimensional transportation problem include the extension of (dTP) to the case of forbidden entries and the d -dimensional bottleneck transportation problem. For the former problem this gives an alternative proof for its greedy solvability in the case of cost arrays which are Monge (another proof follows from the work of Queyranne et al. [105]).

7. Permuted Monge structures

In Sections 3–6 we discussed various classes of optimization problems which become easier if the underlying cost matrix or cost array has a Monge property. Many of these applications are invariant to permutations applied to the given cost structure. For example, in the transportation problem the ordering of the rows and columns of its cost matrix does not play a role. This motivates the following definitions:

An $m \times n$ matrix C is said to be a *permuted Monge matrix* if there exists a pair (ϕ, ψ) of row and column permutations such that the permuted matrix $C_{\phi, \psi} := (c_{\phi(i)\psi(j)})$ becomes a Monge matrix. *Permuted bottleneck Monge matrices* and *permuted algebraic Monge matrices* are defined analogously.

Similarly, an $n_1 \times \cdots \times n_d$ array C is called *permuted (algebraic) Monge array* if there exist permutations ϕ_1, \dots, ϕ_d where ϕ_p acts on the set $\{1, \dots, n_p\}$, $p = 1, \dots, d$, such that the permuted array $C_{\phi_1, \phi_2, \dots, \phi_d}$ with entries $c[\phi_1(i_1), \phi_2(i_2), \dots, \phi_d(i_d)]$ becomes (algebraic) Monge.

In the sequel we will deal with the following general type of recognition problem: Given a matrix (resp. an array) C with elements drawn from a totally ordered commutative semigroup (H, \oplus, \preceq) where \oplus is compatible with \preceq , decide whether or not C belongs to the class of permuted algebraic Monge matrices (resp. arrays) and in the affirmative case, determine a set of permutations which do the transformation job.

7.1. Permuted algebraic Monge matrices in strongly cancellative semigroups

In this subsection we deal with the recognition of permuted algebraic Monge matrices where the underlying algebraic system (H, \oplus, \preceq) satisfies the strong cancellation rule (39).

Deineko and Filonenko [47] present an $O(mn + m \log m + n \log n)$ time algorithm which, given an $m \times n$ matrix C , either finds a pair of permutations (ϕ, ψ) such that $C_{\phi, \psi}$ becomes a Monge matrix or proves that no such permutations exist. Below we show that a slight modification of the main idea in [47] leads to a universal recognition algorithm for the class of permuted algebraic Monge matrices in strongly cancellative semigroups.

In order to facilitate the statement of the recognition algorithm below, we introduce a total order \preceq_I on the set $H \times H$ as follows:

$$(c_1, c_2) \preceq_I (c'_1, c'_2) \iff c_1 \oplus c'_2 \preceq c_2 \oplus c'_1.$$

Algorithm 1. Recognition of permuted algebraic Monge matrices in strongly cancellative semigroups:

1. Find a row r and a column s such that $c_{11} \oplus c_{rs} \neq c_{1s} \oplus c_{r1}$. If no such pair (r, s) exists, then C itself is an algebraic Monge matrix.

2. Determine a permutation $\tilde{\psi}$ of the columns such that

$$(c_{1\tilde{\psi}(1)}, c_{r\tilde{\psi}(1)}) \preceq_I (c_{1\tilde{\psi}(2)}, c_{r\tilde{\psi}(2)}) \preceq_I \dots \preceq_I (c_{1\tilde{\psi}(n)}, c_{r\tilde{\psi}(n)}).$$

3. Find the largest value $p \in \{1, \dots, n\}$ such that

$$c_{1\tilde{\psi}(j)} \oplus c_{r\tilde{\psi}(j+1)} = c_{1\tilde{\psi}(j+1)} \oplus c_{r\tilde{\psi}(j)} \quad \text{for all } j = 1, \dots, p-1$$

and the smallest value $q \in \{1, \dots, n\}$ such that

$$c_{1\tilde{\psi}(j)} \oplus c_{r\tilde{\psi}(j+1)} = c_{1\tilde{\psi}(j+1)} \oplus c_{r\tilde{\psi}(j)} \quad \text{for all } j = q, \dots, n-1.$$

4. Determine rows r_1 resp. r_2 such that

$$\max_{1 \leq k \leq p} (c_{r_1\tilde{\psi}(k)}, c_{i\tilde{\psi}(k)}) \preceq_I \min_{q \leq \ell \leq n} (c_{r_1\tilde{\psi}(\ell)}, c_{i\tilde{\psi}(\ell)}),$$

resp.

$$\max_{1 \leq k \leq p} (c_{i\tilde{\psi}(k)}, c_{r_2\tilde{\psi}(k)}) \preceq_I \min_{q \leq \ell \leq n} (c_{i\tilde{\psi}(\ell)}, c_{r_2\tilde{\psi}(\ell)})$$

holds for all $i = 1, \dots, m$ where “max” and “min” in the lines above are taken w.r.t. the induced order \preceq_I . If no such rows exist, then terminate, since then C is not a permuted algebraic Monge matrix. Otherwise set $\phi(1) := r_1$ and $\phi(m) := r_2$.

5. Find a permutation ψ of the columns such that

$$(c_{\phi(1)\psi(1)}, c_{\phi(m)\psi(1)}) \preceq_I (c_{\phi(1)\psi(2)}, c_{\phi(m)\psi(2)}) \preceq_I \dots \preceq_I (c_{\phi(1)\psi(n)}, c_{\phi(m)\psi(n)}).$$

6. Find the remaining values $\phi(2), \dots, \phi(m-1)$ of the permutation ϕ such that

$$(c_{\phi(1)\psi(1)}, c_{\phi(1)\psi(n)}) \preceq_I (c_{\phi(2)\psi(1)}, c_{\phi(2)\psi(n)}) \preceq_I \dots \preceq_I (c_{\phi(m)\psi(1)}, c_{\phi(m)\psi(n)}).$$

7. Check whether $c_{\phi(i)\psi(j)} \oplus c_{\phi(i+1)\psi(j+1)} \preceq c_{\phi(i)\psi(j+1)} \oplus c_{\phi(i+1)\psi(j)}$ holds for all $1 \leq i \leq m-1$, $1 \leq j \leq n-1$. If this is the case, return the pair of permutations (ϕ, ψ) , otherwise C is not a permuted algebraic Monge matrix.

Theorem 7.1. *Let C be an $m \times n$ matrix with entries drawn from a totally ordered and strongly cancellative semigroup (H, \oplus, \preceq) . Then Algorithm 1 needs at most $O(nm + m \log m + n \log n)$ elementary operations, i.e. comparisons w.r.t. \preceq and \oplus operations, in order to find either a pair (ϕ, ψ) of permutations such that $C_{\phi, \psi}$ is an algebraic Monge matrix or to prove that no such permutations exist.*

Proof. Since the final step checks whether $C_{\phi, \psi}$ is indeed an algebraic Monge matrix, it only remains to be shown that Algorithm 1 does find a pair of permutations (ϕ, ψ) transforming C into an algebraic Monge matrix whenever such a pair exists.

Assume for a while that $\phi(1)$ and $\phi(m)$, i.e. the rows of C which become the first resp. last row of $C_{\phi, \psi}$ are already known. We first proof that a valid permutation ψ is determined in Step 5. This is trivial if the choice of ψ is unique. Thus suppose that there exist two distinct indices ℓ_1 and ℓ_2 such that

$$c_{\phi(1)\psi(\ell_1)} \oplus c_{\phi(m)\psi(\ell_2)} = c_{\phi(1)\psi(\ell_2)} \oplus c_{\phi(m)\psi(\ell_1)}. \quad (57)$$

Since any submatrix of an algebraic Monge matrix has to be algebraic Monge itself, obviously we must have

$$c_{\phi(1)\psi(\ell_1)} \oplus c_{\phi(i)\psi(\ell_2)} \preceq c_{\phi(1)\psi(\ell_2)} \oplus c_{\phi(i)\psi(\ell_1)} \quad \text{and}$$

$$c_{\phi(i)\psi(\ell_1)} \oplus c_{\phi(m)\psi(\ell_2)} \preceq c_{\phi(i)\psi(\ell_2)} \oplus c_{\phi(m)\psi(\ell_1)}$$

for all $i = 2, \dots, m-1$. Together with (57) and the strong cancellation rule this implies

$$c_{\phi(1)\psi(\ell_1)} \oplus c_{\phi(i)\psi(\ell_2)} = c_{\phi(1)\psi(\ell_2)} \oplus c_{\phi(i)\psi(\ell_1)} \quad \text{and}$$

$$c_{\phi(i)\psi(\ell_1)} \oplus c_{\phi(m)\psi(\ell_2)} = c_{\phi(i)\psi(\ell_2)} \oplus c_{\phi(m)\psi(\ell_1)}.$$

Consequently, the mutual position of ℓ_1 and ℓ_2 in ψ does not matter. In a similar way it can be shown that in Step 6, where a row permutation ϕ is determined, ties can be broken arbitrarily, too.

The only remaining problem is how to find $\phi(1)$ and $\phi(m)$. This is done in Steps 2–4. W.l.o.g. we can assume that in the permuted matrix $C_{\phi,\psi}$ row 1 precedes the row r determined in Step 1. This implies that the columns in the set $J_1 := \{\tilde{\psi}(1), \dots, \tilde{\psi}(p)\}$ are candidates for becoming the first column in $C_{\phi,\psi}$, and all columns in the set $J_2 := \{\tilde{\psi}(q), \dots, \tilde{\psi}(n)\}$ are candidates for becoming the last column in $C_{\phi,\psi}$. Hence the following has to hold:

$$c_{\phi(i)k} \oplus c_{\phi(i')\ell} \preceq c_{\phi(i)\ell} \oplus c_{\phi(i')k} \quad \text{for all } 1 \leq i < i' \leq m, k \in J_1 \text{ and } \ell \in J_2,$$

or equivalently,

$$\max_{1 \leq k \leq p} (c_{\phi(i)\tilde{\psi}(k)}, c_{\phi(i')\tilde{\psi}(k)}) \preceq_I \min_{q \leq \ell \leq n} (c_{\phi(i)\tilde{\psi}(\ell)}, c_{\phi(i')\tilde{\psi}(\ell)}) \quad \text{for all } 1 \leq i < i' \leq m.$$

Setting $i = 1$ resp. $i' = m$ yields the decision rules used in Step 4 to determine $\phi(1)$ and $\phi(m)$. This completes the proof of correctness for Algorithm 1.

The complexity analysis is straightforward. Since Steps 2, 5 and 6 involve just sorting, they require in total $O(n \log n + m \log m)$ elementary steps. Steps 1, 4 and 7 can be performed using $O(nm)$ elementary steps whereas $O(n)$ elementary steps suffice for Step 3. \square

In the special case $(\mathbb{R}, +, \leq)$ which leads to Monge matrices some of the steps of Algorithm 1, in particular Step 4, can be performed in a modified form exploiting the fact that $(\mathbb{R}, +)$ forms a group which implies the existence of inverse elements (cf. [47]).

7.2. Permutated algebraic Monge matrices in weakly cancellative semigroups

Algorithm 1 crucially depends on the strong cancellation rule; it does not even apply to the case of semigroups (H, \oplus, \preceq) which fulfill the weak cancellation rule (43). Thus Algorithm 1 can, e.g., be used for the recognition of permuted Monge matrices and permuted strict bottleneck Monge matrices, but it may fail for permuted Monge matrices with infinite entries and for general permuted bottleneck Monge matrices. Nevertheless, these two classes can be recognized in polynomial time as will be discussed below. It is not known, however, whether this polynomiality result carries over to the entire class of permuted algebraic Monge matrices in weakly cancellative semigroups.

In any case we cannot hope for a polynomial time algorithm for recognizing the class of permuted algebraic Monge matrices if the compatibility requirement (37) is the only condition the ordered semigroup (H, \oplus, \preceq) has to fulfill, since this problem has been shown to be NP-hard by Woeginger [134]. The special case where the given matrix C has only two rows or two columns is, however, efficiently solvable, since in that case there always exists a pair of permutations (ϕ, ψ) : just sort the columns resp.

the rows such that the desired Monge property is fulfilled. For the sum case this has already been observed in Dietrich [54] and for the bottleneck case in Klinz et al. [87].

7.2.1. Permuted bottleneck Monge matrices

In the following we will sketch the main ideas of a recognition algorithm which is specially tailored to the case of bottleneck Monge matrices and relies heavily on the threshold characterization of bottleneck Monge matrices given in Observation 4.3. Let again $\tilde{c}_1 > \tilde{c}_2 > \dots > \tilde{c}_L$ denote the sequence of all pairwise distinct values of entries in the input matrix C and associate with each value \tilde{c}_k , $1 \leq k \leq L$, its threshold matrix $T^k = (t_{ij}^k)$ by setting $t_{ij}^k = 0$ if $c_{ij} \leq \tilde{c}_k$ and $t_{ij}^k = 1$ otherwise. Then C is a permuted bottleneck Monge matrix if and only if there exists a common pair (ϕ, ψ) of row and column permutations such that the permuted matrices $T_{\phi, \psi}^k$, $k = 1, \dots, L$, all are bottleneck Monge matrices. To exploit this idea algorithmically, we need an efficient recognition algorithm for the special case of permuted 0–1 bottleneck Monge matrices.

As discussed in Section 4, 0–1 bottleneck Monge matrices correspond to the complement of strongly ordered bipartite permutation graphs. Hence the class of $m \times n$ permuted 0–1 bottleneck Monge matrices can be recognized in $O(nm)$ time by applying the algorithm of Spinrad et al. [118] which decides in linear time whether a given bipartite graph H is a bipartite permutation graph and if so, determines a strong ordering of H .

An alternative linear time algorithm for the recognition of permuted 0–1 bottleneck Monge matrices which is based on the double staircase characterization of 0–1 bottleneck Monge matrices is proposed in Klinz et al. [87].

The recognition problem for permuted bottleneck Monge matrices with arbitrary entries can now be solved by combining the following ideas: Using one of the recognition algorithms for 0–1 matrices mentioned above, we can check for each $k = 1, \dots, L$ whether the corresponding threshold matrix T^k is bottleneck Monge. This is, however, not sufficient since there must exist a pair of permutations (ϕ, ψ) which works for all threshold matrices at the same time. Fortunately, this problem can be solved rather easily by observing that the set $\mathcal{P}(C)$ of all pairs (ϕ, ψ) of row and column permutations which transform a given 0–1 matrix C into a bottleneck Monge matrix can be described in a concise and compact way (cf. Klinz et al. [87]). Therefore, it only remains to either find a pair of permutations (ϕ, ψ) in the intersection of the sets $\mathcal{P}(T^k)$, $k = 1, \dots, L$, or to prove that this intersection is empty. Obviously, the original matrix C is a permuted bottleneck Monge matrix if and only if this intersection is nonempty.

From the point of view of efficiency it turns out to be advantageous not to compute the full intersection $\mathcal{P}' := \bigcap_{k=1}^L \mathcal{P}(T^k)$, but only a subset $\tilde{\mathcal{P}'}$ with the property that if this subset is empty also \mathcal{P}' is empty. The set $\tilde{\mathcal{P}'}$ is built up incrementally in a step-by-step fashion whenever a new threshold matrix has been investigated. A direct implementation of this idea leads to an $O(Lnm)$ time algorithm which is $O(n^2m^2)$ in the worst case. A closer investigation shows, however, that it is not necessary to ex-

plore all threshold values since only at most $O(n+m)$ of these provide a new piece of information about the structure of the intersection \tilde{P}' . This leads to an $O((n+m)nm)$ time recognition algorithm for the class of $m \times n$ bottleneck Monge matrices. Further details can be found in Klinz et al. [87].

7.2.2. Permuted Monge matrices with infinite entries

Given an $m \times n$ matrix C with entries $c_{ij} \in \mathbb{R} \cup \{\infty\}$ as input, the following two-stage algorithm either finds a pair of permutations (ϕ, ψ) such that $C_{\phi, \psi}$ is a Monge matrix with infinite entries or proves that no such permutations exist.

In the first stage we associate with C a skeleton matrix T which is defined by setting $t_{ij} := 0$ whenever c_{ij} is finite and $t_{ij} := 1$ otherwise. It can easily be checked that if C is a permuted Monge matrix, then T must necessarily be a permuted bottleneck Monge matrix. Hence, we apply a recognition algorithm for permuted bottleneck Monge matrices to T and terminate if this test fails. Otherwise, the first stage delivers a pair of permutations $(\tilde{\phi}, \tilde{\psi})$ such that $T_{\tilde{\phi}, \tilde{\psi}}$ is bottleneck Monge. In this case we continue with the second stage and apply Algorithm 1 to the matrix $C_{\tilde{\phi}, \tilde{\psi}}$ in order to determine the mutual placement of those rows and columns whose positions are not already implied from the results of the first stage. In fact, it suffices to apply Algorithm 1 to those blocks of $C_{\tilde{\phi}, \tilde{\psi}}$ which do not contain infinite entries. Overall we get an algorithm which runs in $O(nm + m \log m + n \log n)$ time.

7.3. Permuted Monge arrays

Permuted Monge arrays can be recognized by exploiting the characterization of Monge arrays in terms of its two-dimensional subarrays (cf. Lemma 6.3). Combining this observation with the algorithm of Deineko and Filonenko [47] for recognizing permuted Monge matrices, Rudolf [111] obtained an algorithm which decides in $O(d^2 n_2 n_3 \cdots n_d (n_1 + \log n_d))$ time whether or not a given $n_1 \times n_2 \times \cdots \times n_d$ array C with $n_1 \leq n_2 \leq \cdots \leq n_d$ is a permuted Monge array. In the affirmative case, the algorithm furthermore delivers permutations ϕ_1, \dots, ϕ_d such that $C_{\phi_1, \dots, \phi_d}$ becomes a Monge array.

7.4. Permuted algebraic Monge arrays

Permuted algebraic Monge arrays in strongly cancellative semigroups (H, \oplus, \preceq) can be recognized in nearly the same way as described above for the class of permuted Monge arrays. We only have to replace the algorithm of Deineko and Filonenko by Algorithm 1. This approach works because the strong cancellation rule ensures the essential property that a d -dimensional array is algebraic Monge if and only if all of its two-dimensional subarrays are algebraic Monge matrices.

Unlike in the two-dimensional case, nothing at all is, however, known for the recognition of permuted algebraic Monge matrices in weakly cancellative semigroups. Even the special case of recognizing permuted bottleneck Monge arrays is open.

7.5. Restricted recognition problems

In some applications an arbitrary pair (ϕ, ψ) of row and column permutations which transforms a given matrix C into a matrix $C_{\phi, \psi}$ satisfying a specific Monge property is of no use. In these cases we have to pose additional restrictions on ϕ and ψ .

For example, in problems where the rows and columns of C may not be permuted independently without changing the problem (take e.g. the travelling salesman problem), we have to require that $\phi = \psi$. There also exist problems where we are not allowed to permute the rows or the columns: take, e.g., the flow shop scheduling problem discussed in Section 4 where we are not allowed to change the fixed order according to which the jobs have to pass the machines since this order is part of the input. In yet another class of restricted recognition problems we want to find ϕ and ψ such that the permuted matrix has some additional properties, e.g. we might require that the rows or the columns are monotone increasing (or monotone decreasing). Obviously, in higher dimension there are even more possibilities for additional restrictions on the permutations applied to an array C in order to transform it into a Monge array.

Let us briefly mention what is known about restricted recognition problems of the type mentioned above.

For the restricted problem where $\phi = \psi$ has to hold, it is most important to have available a complete characterization of the set of all pairs (ϕ, ψ) of row and column permutations such that the permuted matrix $C_{\phi, \psi}$ has the desired Monge property. For the sum case such a characterization is given in Rudolf [111] and for the bottleneck case in Klinz et al. [87]. On the basis of these characterizations, algorithms can be developed which, given an $n \times n$ matrix C , either find a permutation ϕ such that $C_{\phi, \phi}$ is (bottleneck) Monge or determine that no such permutation exists. The algorithm for the sum case runs in $O(n^2)$ time and the algorithm for the bottleneck case in $O(n^3)$ time. The first algorithm can easily be extended to yield an $O(n^2)$ algorithm for permuted algebraic Monge matrices in strongly cancellative semigroups.

The restriction that the row permutation ϕ has to be the identity permutation usually makes the underlying recognition problem easier. A straightforward adaptation of Algorithm 1 solves this problem for the sum case and for the general algebraic case in strongly cancellative semigroups. The running time of the resulting algorithm is $O(mn + n \log n)$. For the bottleneck case an $O(\min\{n^2m, nm^2 \log n\})$ time algorithm is derived in [87].

To conclude this section, let us mention that the recognition algorithms mentioned above can be modified in such a way that they construct permutations such that all rows and/or all columns are monotone in the same direction. The time complexities remain the same.

8. Monge Sequences

8.1. Definitions and fundamental properties of Monge sequences

In the course of his investigations on greedily solvable transportation problems Hoffman [75] introduced the concept of Monge sequences. Given an $m \times n$ real matrix C , an ordering $\mathcal{S} = ((i_1, j_1), \dots, (i_{mn}, j_{mn}))$ of the nm pairs of indices of C is called *Monge sequence* (with respect to the matrix C) if the following condition holds:

For every $1 \leq i, r \leq m$ and $1 \leq j, s \leq n$, whenever (i, j) precedes both (i, s) and (r, j) in \mathcal{S} , the corresponding matrix entries in C are such that

$$c_{ij} + c_{rs} \leq c_{is} + c_{rj}. \quad (58)$$

If there exist permutations π and ρ such that the pair $(\pi(i), \rho(j))$ precedes the pair $(\pi(r), \rho(s))$ in \mathcal{S} whenever $(\pi(i), \rho(j))$ is lexicographically smaller than $(\pi(r), \rho(s))$, then the Monge sequence \mathcal{S} is said to be a *lexicographic Monge sequence*.

Note that any permuted Monge matrix C has a Monge sequence (take e.g. the lexicographic Monge sequence $\mathcal{S} = ((\phi(1), \psi(1)), (\phi(1), \psi(2)), \dots, (\phi(1), \psi(n)), (\phi(2), \psi(1)), \dots, (\phi(m), \psi(n)))$ where ϕ, ψ are such that $C_{\phi, \psi}$ is Monge), but not every matrix which has a Monge sequence is also a permuted Monge matrix, take, e.g. the matrix

$$\begin{pmatrix} 1 & 8 & 12 \\ 10 & 16 & 5 \\ 18 & 4 & 14 \end{pmatrix}.$$

In other words, not every matrix which has a Monge sequence also has a lexicographic one.

The definition (58) above can be extended to cover Monge sequences for matrices with infinite (forbidden) entries by just extending $+$ and \leq to $\mathbb{R} \cup \{\infty\}$ in the usual way (see [54, 116, 117]). Note that as in the case of Monge matrices, infinite entries cannot be simulated by a sufficiently large real number without possibly destroying the Monge property.

Dietrich [54] observed that Monge sequences and Monge sequences with forbidden entries can also be interpreted in terms of a special antimatroid, the so-called *Monge antimatroid*. Antimatroids form a subclass of greedoids, see Korte et al. [88]. This interpretation leads to a deeper understanding of the structure of Monge sequences.

For the special case of 0-1 matrices Hoffman [76] gave the following nice combinatorial characterization of the subclass of matrices having a Monge sequence.

Theorem 8.1. *Let C be an $m \times n$ 0-1 matrix and let \bar{C} denote its 0-1 complement, i.e. the matrix which is obtained from C by exchanging the role of zeros and ones. C has a Monge sequence if and only if \bar{C} is totally balanced, or equivalently, if and only if the rows and columns of \bar{C} can be permuted such that the permuted matrix does not contain the 2×2 matrix $\Gamma = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ as a submatrix.*

Combining Theorem 8.1 with a result of Spinrad [119] which states that there exist $O(2^{\theta(n \log^2 n)})$ totally balanced $n \times n$ matrices gives a feeling for the cardinality of the set of 0–1 matrices having a Monge sequence. We are not aware of similar results for matrices with arbitrary entries.

The concept of Monge sequences can be generalized in two directions. First, it is straightforward to define *algebraic Monge sequences* by proceeding analogously as in Section 5. Again the entries of the matrix C are taken from a totally ordered semigroup (H, \oplus, \preceq) which fulfills property (37) and $+$ and \leqslant in (58) are replaced by \oplus and \preceq , respectively.

More difficult is, however, the step from Monge sequences for matrices to d -dimensional Monge sequences with respect to d -dimensional arrays for $d \geq 3$. We first need some notation. Let C be an $n_1 \times n_2 \times \cdots \times n_d$ array and define $N_k := \{1, 2, \dots, n_k\}$, $k = 1, \dots, d$. \mathcal{S} is called a *d -dimensional sequence* w.r.t. the array C if it is an ordering of the elements in the Cartesian product $\mathcal{N} = N_1 \times \cdots \times N_d$. Furthermore, let $\mathcal{I} := \{(i_1^k, i_2^k, \dots, i_d^k) \mid i_\ell^k \in N_\ell, 1 \leq \ell \leq d, 1 \leq k \leq q\}$ be a subset of \mathcal{N} of cardinality q and define $\mathcal{L}_\ell(\mathcal{I}) := \{i_\ell^1, \dots, i_\ell^q\}$ as the multiset (list) of all integers which occur as ℓ th coordinate of an element in \mathcal{I} .

A set $\mathcal{I} \subseteq \mathcal{N}$ is said to be *feasible* with respect to the d -tuple $(i_1, i_2, \dots, i_d) \in \mathcal{N}$ if and only if for all $k = 1, \dots, q$ the entry i_k is contained at least once in the multiset $\mathcal{L}_k(\mathcal{I})$. If \mathcal{I} is feasible but no proper subset of \mathcal{I} is feasible, then \mathcal{I} is said to be *minimal*. Finally, let $M(\mathcal{I}) := \{(i_1, i_2, \dots, i_d) \mid i_k \in \mathcal{L}_k(\mathcal{I})\}$, i.e. $M(\mathcal{I})$ contains all d -tuples $(i_1, \dots, i_d) \in \mathcal{N}$ which can be formed by choosing the k th coordinate i_k from the multiset $\mathcal{L}_k(\mathcal{I})$ for all $k = 1, \dots, d$.

Now we are prepared to introduce the notion of a d -dimensional Monge sequence. A sequence \mathcal{S} is called a *d -dimensional Monge sequence* w.r.t. the array C if the following condition is satisfied (cf. [112]):

Let $(i_1^1, i_2^1, \dots, i_d^1) \in \mathcal{S}$. Then for each set $\mathcal{I} := \{(i_1^k, i_2^k, \dots, i_d^k) \mid i_\ell^k \in N_\ell, 1 \leq \ell \leq d, 1 \leq k \leq q\}$ which is minimal with respect to $(i_1^1, i_2^1, \dots, i_d^1)$ we need to have that whenever $(i_1^1, i_2^1, \dots, i_d^1)$ is the element which occurs first in \mathcal{S} among all elements of $M(\mathcal{I})$ then

$$c[i_1^1, i_2^1, \dots, i_d^1] + \min_{\phi_2, \dots, \phi_d} \left\{ \sum_{k=2}^q c[i_1^k, i_2^{\phi_2(k)}, \dots, i_d^{\phi_d(k)}] \right\} \leq \sum_{(j_1, j_2, \dots, j_d) \in \mathcal{I}} c[j_1, j_2, \dots, j_d] \quad (59)$$

has to hold, where ϕ_2, \dots, ϕ_d are bijections acting on the set $\{2, \dots, q\}$.

In a similar way *d -dimensional algebraic Monge sequences* can be introduced by replacing $+$ and \leqslant by \oplus and \preceq .

8.2. Applications of Monge sequences

Hoffman [75] introduced Monge sequences because matrices with a Monge sequence lead to a larger class of greedily solvable transportation problems. More specifically, given a sequence $\mathcal{S} = ((i_1, j_1), \dots, (i_m, j_m))$ consider the following greedy algorithm

Greedy(\mathcal{S}): Take the elements of \mathcal{S} in this order and maximize the corresponding variables in turn, i.e. in the k th step of Greedy(\mathcal{S}) set the variable $x_{i_k j_k}$ to the largest value such that the partial solution $(x_{i_1 j_1}, \dots, x_{i_{k-1} j_{k-1}}, x_{i_k j_k})$ does not violate the transportation constraints (17)–(19).

Obviously, the algorithm Greedy(\mathcal{S}) not only applies to the Hitchcock transportation problem (TP) but also to the bottleneck transportation problem and more generally to the algebraic transportation problem (ATP). If the cost matrix C contains no infinite entries, then Greedy(\mathcal{S}) always determines a feasible transportation schedule.

The theorem below gives a condition on the sequence \mathcal{S} and the cost matrix C which guarantees that the solution found by Greedy(\mathcal{S}) is also optimal. For the case of the (TP) and cost matrices with real entries only, this theorem has already been obtained by Hoffman [75], while the extension to the (TP) with cost matrices containing infinite entries is due to Shamir and Dietrich [54, 116, 117]. More generally, this result can be formulated for the algebraic transportation problem (ATP) and algebraic Monge sequences.

Theorem 8.2. *The algorithm Greedy(\mathcal{S}) constructs an optimal solution of the algebraic transportation problem (ATP) with cost matrix C for all feasible supply and demand vectors if and only if \mathcal{S} is an algebraic Monge sequence with respect to the matrix C .*

Theorem 8.2 generalizes Theorem 5.2 in Section 5, which covers only the case of lexicographic algebraic Monge sequences, and can be proved in an analogous way.

The main disadvantage of the greedy algorithm Greedy(\mathcal{S}) is that for sequences \mathcal{S} which are not lexicographic, the solution (x_{ij}) determined by Greedy(\mathcal{S}) is in general not a basic solution of the transportation problem. Hence it may be necessary to scan the whole sequence \mathcal{S} which takes $O(nm)$ time in the worst case, while for lexicographic sequences, as in the case of the north-west corner rule, only $O(n + m)$ steps are required (in each step at least one node is eliminated). It is an interesting open problem whether there exists a method for preprocessing the sequence \mathcal{S} such that the greedy algorithm when applied to the preprocessed sequence will take only $O(n + m)$ steps for any supply and demand vectors. This would be useful if the cost matrix C remains the same, but the supplies and demands change, and the problem has to be resolved according to the new data.

The related problem of modifying Greedy(\mathcal{S}) such that it constructs a basic solution can, however, be solved by combining the greedy algorithm with a vertex elimination procedure as shown by Adler and Shamir [2]. This algorithm still takes $O(nm)$ time, but may be useful in performing post-optimality analysis and in the construction of optimal solutions to the dual of the transportation problem.

As one would expect from the results reported so far, there exists a multidimensional version of Theorem 8.2. Let $\text{Greedy}_d(\mathcal{S})$ denote the d -dimensional extension of the greedy algorithm Greedy(\mathcal{S}) introduced above. $\text{Greedy}_d(\mathcal{S})$ again takes the

index tuples (i_1, \dots, i_d) in the order they occur in \mathcal{S} and maximizes the corresponding variables x_{i_1, \dots, i_d} in turn.

Theorem 8.3 (Rudolf [112]). *Algorithm Greedy_d(\mathcal{S}) constructs an optimal solution of the d -dimensional axial transportation problem for arbitrary right-hand side vectors if and only if \mathcal{S} is a d -dimensional Monge sequence with respect to the cost array C .*

Again this result can easily be extended to the d -dimensional axial algebraic transportation problem.

The previous results only regard optimality aspects of the greedy algorithm. If the cost matrix (array) C contains, however, infinite entries as well, then Greedy_d(\mathcal{S}) may construct a solution which is not even feasible. Therefore, Adler et al. [1] investigated conditions on the sequence \mathcal{S} and the cost matrix C under which Greedy(\mathcal{S}) always produces a feasible solution. To that end, they introduced the concept of a so-called *feasibility sequence* and proved a one-to-one relationship between the feasibility sequences of a matrix C and the Monge sequences of the matrix C' defined by $c'_{ij} = 1$ if c_{ij} is finite and $c'_{ij} = \infty$ otherwise. Instead of C' one can also use the matrix \widehat{C} where \widehat{c}_{ij} is one if c_{ij} is finite and zero otherwise. This relates feasibility sequences with totally balanced matrices (cf. Theorem 8.1) and hence with chordal bipartite graphs, i.e. bipartite graphs which do not contain an induced chordless cycle of length six or more (cf. Golumbic [68] and Adler et al. [1]). It is not difficult to show that the concept of feasibility sequences directly carries over to d -dimensional cost arrays for $d \geq 3$.

8.3. Construction and detection of Monge sequences

In this subsection we describe a simple algorithm for the detection of Monge sequences which is due to Alon et al. [14]. This algorithm can also be applied to finding algebraic Monge sequences.

Recall that each inequality in the condition (58) such that $i \neq r$ and $j \neq s$ involves four matrix entries which form a 2×2 submatrix. This submatrix has two diagonals, one containing the entries c_{ij} and c_{rs} and the other containing the entries c_{is} and c_{rj} . If $c_{ij} + c_{rs} < c_{is} + c_{rj}$ holds, then the diagonal corresponding to the index pairs (i, j) and (r, s) is said to be a *small diagonal* and the diagonal corresponding to the index pairs (i, s) and (r, j) is said to be *large diagonal*.

Obviously, an element (i, j) which lies on a large diagonal can never occur as the first element in a Monge sequence. Based on this idea Alon et al. [14] derived the following algorithm.

Algorithm 2. Construction of Monge sequences

1. Build a graph G whose nodes correspond to the set of all index pairs (i, j) , $1 \leq i \leq m$ and $1 \leq j \leq n$. Connect two nodes iff the corresponding index pairs belong to a large diagonal.

2. As long as there is an isolated node (i, j) in G , place (i, j) next in the sequence and delete it. Additionally eliminate all edges in G which connect a node (i, s) with a node (r, j) .

3. If finally G is empty, we have constructed a Monge sequence, otherwise no Monge sequence exists.

A straightforward implementation of Algorithm 2 performs $O(m^2n^2)$ elementary steps (\oplus operations and comparisons w.r.t. \preceq) and requires $O(mn)$ space. Alon et al. [14] considered only the classical case of Monge sequences where $(H, \oplus, \preceq) := (\mathbb{R}, +, \leq)$. For this case they could improve the running time of Algorithm 2 to $O(m^2n \log n)$ at the expense of increasing the space requirement to $O(m^2n)$ where m can be assumed to be the smaller of the dimensions n and m (otherwise take the transpose of C). It is easy to see that this approach works for the general case of algebraic Monge sequences in strongly cancellative semigroups. Combining the techniques of Alon et al. with Lemma 4.2 an $O(m^2n \log n)$ algorithm can also be achieved for the detection of bottleneck Monge sequences though in that case the strong cancellation rule does not hold (cf. [87] and [110]).

For the detection of Monge sequences in matrices with a small number e of finite entries Algorithm 2 is not very efficient. Improved algorithms for this case which take $O(emn)$ time and $O(mn)$ space resp. $O(em \log n)$ time and $O(m^2n)$ space are given by Shamir and Dietrich [116, 117].

For $d \geq 3$ only the detection of d -dimensional Monge sequences has been investigated. Rudolf [112] generalized Algorithm 2 and obtained an algorithm which identifies a d -dimensional Monge sequence in an $n \times n \times \cdots \times n$ array C in $O(n^{d^2})$ time whenever such a sequence exists.

9. Other Monge-like properties

In this section we give a brief survey of other Monge-like properties which have been discussed in the literature but which did not fit into the framework of the previous sections. We start with a kind of *continuous* Monge property and afterwards investigate Monge properties which are obtained by relaxing the classical Monge property (2).

9.1. Continuous Monge property

A function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ is said to satisfy the *continuous Monge property* if it is submodular w.r.t. the lattice $(\mathbb{R}^2, \vee, \wedge)$ where \vee resp. \wedge are again defined as the componentwise maximum resp. componentwise minimum (cf. Section 2), or equivalently if

$$f(x, y) + f(x', y') \leq f(x', y) + f(x, y') \quad \text{for all } x \leq x', y \leq y'. \quad (60)$$

Similarly, $f : \mathbb{R}^k \mapsto \mathbb{R}$ is said to satisfy the *k-dimensional continuous Monge property* if it is submodular w.r.t. $(\mathbb{R}^k, \vee, \wedge)$, or equivalently if f is submodular (i.e.

satisfies (60)) with respect to any two of its arguments.

Obviously, the continuous Monge property can again be embedded into an algebraic framework. But, at least to the authors' knowledge such generalizations have not been investigated so far. The only exception is the case where $k = 2$ and the addition is replaced by the multiplication and the order is reversed which leads to the notion of total positivity of order 2, cf. Karlin [83].

Functions with the continuous Monge property already arose as generating functions for Monge matrices and Monge arrays in Sections 2 and 6. This class of functions plays, however, a much larger role in various branches of mathematics as might be suspected. In the following we will only deal with applications which are in some sense related to the discrete applications discussed in previous sections.

Applications of the continuous Monge property. Let (U_i, d) , $i = 1, \dots, k$, be separable metric spaces and let F_i , $i = 1, \dots, k$, be nonnegative Borel measures on (U_i, d) such that $F_1(U_1) = F_2(U_2) = \dots = F_k(U_k)$. Furthermore, let \mathcal{U} denote the Cartesian product $U_1 \times U_2 \times \dots \times U_k$ and let $\mathcal{F}(F_1, \dots, F_k)$ denote the space of all nonnegative Borel measures F on \mathcal{U} whose projection to the i th coordinate is equal to F_i , $i = 1, \dots, k$.

Given a continuous cost function $c : \mathcal{U} \mapsto \mathbb{R}$ the *continuous transportation problem* (CTP) can be formulated as follows:

$$\inf \left\{ \int c \, dF : = \int_{\mathcal{U}} c(x_1, \dots, x_k) F(dx_1, \dots, dx_k) : F \in \mathcal{F}(F_1, \dots, F_k) \right\}. \quad (61)$$

The (CTP) belongs to the class of linear programming problems in infinite-dimensional spaces (cf. [15]). By choosing appropriate discrete measures F_i , the (CTP) turns into the k -dimensional axial transportation problem discussed in Section 6 and for $k=2$ we obtain the classical Hitchcock transportation problem treated in Section 3.

Apparently, Kantorovich [81, 82] was the first who dealt with the (CTP) in an abstract mathematical formulation at least for $k=2$ (cf. also [106, 107]). It is interesting to note, however, that the transportation problem considered by Monge [96] in 1781 is a special case of the (CTP) introduced above. In this case we have $k = 2$, $U_i \subseteq \mathbb{R}^2$ or $U_i \subseteq \mathbb{R}^3$ for $i=1, 2$ and $c(x, y) = d(x, y)$ where $d(x, y)$ is the usual Euclidean metric. The measures F_1 and F_2 can be regarded as probability measures which describe the initial and the final distribution of mass and the space $\mathcal{F}(F_1, F_2)$ represents all admissible transference plans F . The quantity $F(dx, dy)$ is then proportional to the amount of mass shipped from a neighbourhood dx of point x to a neighbourhood dy of point y . Hence the objective function in (CTP) measures the total movement of mass which is to be minimized.

For these historical reasons, problems of the type (CTP) are in the literature usually referred to as *Monge–Kantorovich (mass transfer) problems* (MKPs) (or sometimes also as *Kantorovich–Rubinstein problems*). The MKP arises in many areas of mathematics, e.g. in differential geometry, functional analysis, linear programming, probability theory, mathematical statistics, information theory, statistical physics, dynamical

systems and matrix theory (for further details see [107]).

The following theorem of Tchen [122] is a continuous analogue to Theorem 6.5.

Theorem 9.1. *Let $U_i = \mathbb{R}$, $i = 1, \dots, k$ and let F_1, \dots, F_k be distribution functions on \mathbb{R} . Furthermore, suppose that at least one of the following two regularity conditions holds for the cost function $c : \mathbb{R}^k \mapsto \mathbb{R}$ in (CTP):*

- (i) $\sup \left\{ \left| \int_{\mathbb{R}^k} c dF \right| : F \in \mathcal{F}(F_1, \dots, F_k) \right\} < \infty$.
- (ii) $c \geq h$ for some continuous function h such that $\int_{\mathbb{R}^k} h dF$ is finite and constant for all $F \in \mathcal{F}(F_1, \dots, F_k)$.

If c satisfies the k -dimensional continuous Monge property, then an optimal transportation plan $F^* \in \mathcal{F}(F_1, \dots, F_k)$ is given by the so-called Hoeffding distribution

$$F^*(x_1, \dots, x_k) := \min\{F_1(x_1), \dots, F_k(x_k)\} \quad \text{for all } (x_1, \dots, x_k) \in \mathbb{R}^k \quad (62)$$

and the optimal value of (CTP) is given by $\int_{\mathbb{R}^k} c dF^*$.

A slightly weaker result which requires stronger regularity conditions has already been proved by Lorentz [91], but Tchen's proof is simpler. For the special case of $k = 2$ Theorem 9.1 has also been obtained by Cambanis, Simons and Stout [42].

Theorem 9.1 generalizes many results of this type in the literature, both for the discrete case and the continuous case. For example, the famous rearrangement theorems of Hardy, Littlewood and Pólya [72] (see e.g. Theorems 368 and 378) and the classical results of Hoeffding [74] and Fréchet [62, 63] on bivariate distributions with fixed marginals follow from Tchen's theorem. For a brief overview of these connections and many additional references the interested reader is referred to the bibliographical notes in Rachev [107].

In view of Theorem 9.1 the greedy-type results of Balinski and Rachev [17] and Hoffman [75] (cf. Sections 3 and 6) for the discrete transportation problem can be seen as explicit rules for calculating the distribution H^* in the discrete case.

As one might suspect, there also exist continuous analogues to some of the transportation related applications discussed in Section 3. It is beyond the scope of this subsection to give a full account of the work in this field. We just mention that Olkin and Rachev [98, 99] considered the (CTP) with additional constraints on the support of the feasible transportation plans and capacity constraints (cf. Section 3, [18] and [78] for similar discrete problems). Moreover, Balinski and Rachev [17] and Rachev and Rüschendorf [108] consider variations of the (CTP) with other constraints on the given marginals, such as continuous analogues of the inequality constraints discussed in Section 3 and constraints on the sum of the marginals.

9.2. Weak Monge matrices and bisimplicial elimination sequences

Let C be an $m \times n$ matrix. In some applications it suffices to require that C satisfies the following relaxation of the Monge property (2):

$$c_{ii} + c_{rs} \leq c_{is} + c_{ri} \quad \text{for all } 1 \leq i < r \leq m, 1 \leq i < s \leq n. \quad (63)$$

In this paper matrices fulfilling (63) will be referred to as *weak Monge matrices*, but unfortunately there is no consensus on this term in the literature. Some authors (see e.g. [29, 43]) use instead the term *Monge matrix* and attach the name *strong Monge matrix* to those matrices which are called Monge matrices in this paper.

The concept of weak Monge matrices can again be embedded into an algebraic setting which leads to *weak algebraic Monge matrices*. In accordance with the terminology introduced in Section 7 a matrix C is called a *permuted weak (algebraic) Monge matrix* if there exist permutations ϕ and ψ such that $C_{\phi,\psi}$ is a weak (algebraic) Monge matrix.

It is interesting to note that the class of permuted weak Monge matrices can be introduced in the following alternative way: Let $G = (V_1, V_2; E)$ be a bipartite graph with $V_1 = \{1, \dots, m\}$ and $V_2 = \{1, \dots, n\}$ and let $C = (c_{ij})$ denote its weighted adjacency matrix where $c_{ij} = \infty$ if $(i, j) \notin E$.

Following Derigs et al. [53] we say that the edge $(i, j) \in E$ fulfills the *Monge property* with respect to the graph G and the matrix C if

$$c_{ij} + c_{rs} \leq c_{is} + c_{rj} \quad \text{for all } r \in V_1, s \in V_2. \quad (64)$$

Derigs et al. investigate matrices C such that the edge set E of the graph G can be decomposed by iteratively removing the vertices i and j of an edge (i, j) satisfying (64) and all incident edges. More precisely, they study sequences of edges $\mathcal{S} = ((i_1, j_1), \dots, (i_q, j_q))$ of G such that for all $p = 1, \dots, q$ the edge (i_p, j_p) fulfills (64) w.r.t. the graph G_p where $G_1 := G$ and G_p , $p \geq 2$, is obtained from G by deleting all nodes which are incident with edges from the set $\{(i_1, j_1), \dots, (i_{p-1}, j_{p-1})\}$ and removing all edges which are incident with nodes which are deleted in this way. If the final graph G_{q+1} contains no edges, the sequence \mathcal{S} is said to be *complete*.

Note that the elimination process above bears some resemblance with Hoffman's notion of a Monge sequence (cf. Section 8). Hoffman's Monge sequence can also be viewed as kind of elimination sequence, but in his case in the p th step of the elimination process only the edge (i_p, j_p) itself is eliminated. Nevertheless, Derigs et al. [53] used the term Monge sequence for a complete elimination sequence of the type described above, but we do not adopt this convention here and follow instead Faigle [58], who used the term *complete bisimplicial elimination sequence (scheme)*. This choice can be motivated as follows.

Let C' denote the 0–1 matrix which is associated with the given bipartite graph $G = (V_1, V_2; E)$ by setting $c'_{ij} = 0$ if $(i, j) \in E$ and $c'_{ij} = 1$ otherwise. Then an edge $(i, j) \in E$ which fulfills the Monge property (64) w.r.t. the matrix C' corresponds to a *bisimplicial edge* of G as introduced by Golumbic and Goss [69], i.e. an edge (i, j) such that the

bipartite subgraph induced by the neighbours of i and j is complete. Hence there exists a 1–1 correspondence between 0–1 matrices which allow a complete elimination sequence in the sense of Derigs et al. and bipartite graphs for which there exists a *perfect edge elimination scheme*, i.e. an edge elimination scheme which transforms G into a graph with no edges by iteratively removing bisimplicial edges (cf. [69]).

As was the case for Hoffman's Monge sequences, also bisimplicial elimination sequences are closely related to greedoids (cf. [58, 88]).

It can now easily be seen that the class of permuted weak Monge matrices and the class of matrices which allow a complete bisimplicial elimination scheme coincide. More precisely, we have

Observation 9.2. *Given an $m \times n$ matrix C , there exist permutations ϕ and ψ such that $C_{\phi,\psi}$ becomes a weak Monge matrix if and only if there exists a complete bisimplicial elimination sequence $\mathcal{S} = ((i_1, j_1), \dots, (i_q, j_q))$ with respect to C .*

The sequence \mathcal{S} and the permutations ϕ and ψ correspond to each other by the relations $\phi(k) = i_k$ and $\psi(k) = j_k$, $k = 1, \dots, q$. If this rule leaves any values of ϕ or ψ unspecified, ϕ resp. ψ can be arbitrarily augmented to permutations of $\{1, \dots, m\}$ resp. $\{1, \dots, n\}$.

The same idea can be used to show that any matrix C which has a Monge sequence in the sense of Hoffman is a permuted weak Monge matrix. The converse is, however, not true as can be seen from the matrix

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

which fulfills condition (63) but has no Monge sequence (this follows from Theorem 8.1 since C contains in the upper-right corner a 3×3 submatrix whose 0–1 complement is well-known to be not totally balanced). When rephrased in terms of bipartite graphs, we obtain the result of Golumbic and Goss [69] that for every chordal bipartite graph (cf. Section 8) there exists a perfect edge elimination sequence, while the converse is false.

Recently, Fortin and Rudolf [61] generalized the concept of weak Monge matrices to $d \geq 3$ dimensions and introduced d -dimensional weak Monge arrays which form a superclass of the class of d -dimensional Monge arrays introduced in Section 6.

Recognition of weak Monge matrices. When the question comes to the recognition of permuted weak algebraic Monge matrices, first note that there exists a trivial $O(\min\{n, m\}n^2m^2)$ time algorithm to solve this problem. This algorithm successively tries to find an edge (i, j) satisfying (64) which takes $O(n^2m^2)$ steps by checking all $O(mn)$ candidate edges and then eliminates the nodes i and j and all edges which are incident with i or j . This process terminates after at most $\min\{n, m\}$ elimination steps

when either a complete elimination sequence has been obtained or no further edge with the Monge property can be found (then C is not a permuted weak Monge matrix).

A more efficient algorithm for the recognition of $n \times n$ permuted weak Monge matrices which runs in $O(n^4)$ time is given in Cechlárová and Szabó [43]. The authors mention that an improvement to $O(n^3 \log n)$ is possible but do not give further details in their paper.

A much simpler alternative algorithm with the same running time for square matrices follows from the following minor adaptation of the algorithm of Alon et al. [14] for the detection of Monge sequences (cf. Algorithm 2 in Section 8). Instead of deleting only the isolated node (i,j) placed next in the elimination sequence and the edges linking nodes (i,s) with nodes (r,j) we also delete all nodes (i,s) and (r,j) and all incident edges. A straightforward implementation of this method runs in $O(n^2m^2)$ time; using the methods described in Alon et al. [14] the running time can be improved to $O(m^2n \log n)$ time (assume w.l.o.g. that $m \leq n$).

Since Algorithm 2 also works in an algebraic setting, the class of permuted weak algebraic matrices can be recognized in $O(n^2m^2)$ steps. For the recognition of permuted weak Monge matrices with infinite entries, permuted weak bottleneck Monge matrices and permuted weak algebraic Monge matrices in strongly cancellative semigroups the number of performed steps can again be brought down to $O(m^2n \log n)$ (where $m \leq n$).

The recognition problem for permuted weak (algebraic) Monge arrays can be solved by a modification of the ideas in Algorithm 2 for the detection of a Monge sequence. For a given d -dimensional array C of order $n \times n \times \dots \times n$, the algorithm of Fortin and Rudolf [61] runs in $O(d^2(d-1)!^dn^{d^2})$ time.

Applications of weak Monge matrices. While Monge matrices turned out to be essential in characterizing the greedy solvability of transportation problems, weak Monge matrices play an analogous role for assignment problems. For the linear sum assignment problem the theorem below follows immediately from Theorem 2.2 in Derigs et al. [53] and Observation 9.2. The extension to the general algebraic assignment problem is straightforward.

Theorem 9.3. *If the cost matrix C of the linear (algebraic) assignment problem is a weak (algebraic) Monge matrix, then the identity permutation is an optimal assignment.*

Proof. Assume that the identity permutation is not optimal and choose ϕ among all optimal solutions such that $i := \min\{j \in \{1, \dots, n\} : \phi(j) \neq j\}$ is maximum. Then there exist indices $r > i$ and $s > i$ such that $\phi(r) = i$ and $\phi(i) = s$. Since C is a weak algebraic Monge matrix we must have $c_{ii} \oplus c_{rs} \preceq c_{is} \oplus c_{ri}$. Thus the permutation ϕ' with $\phi'(i) := i$, $\phi'(r) := s$ and $\phi'(k) = \phi(k)$ otherwise, is also an optimal assignment. But this contradicts the choice of i and hence we are done. \square

Theorem 9.3 can be generalized to the d -dimensional axial assignment problem in a straightforward way by replacing weak Monge matrices by weak Monge arrays (for

details the reader is referred to Fortin and Rudolf [61]).

Based on the concept of bisimplicial elimination sequences and on Theorem 9.3, Derigs et al. [53] furthermore derived a purely combinatorial optimality criterion for assignment problems and new assignment algorithms based on this criterion. Related results for the maximum cardinality matching problem can be found in [52].

9.3. Supnick matrices

Another class of Matrices which are closely related to Monge matrices are *Supnick matrices* which are defined as follows: A symmetric $n \times n$ matrix C is called a *Supnick matrix* if

$$c_{ij} + c_{rs} \leq c_{ir} + c_{js} \leq c_{is} + c_{jr} \quad \text{for } 1 \leq i < j < r < s \leq n. \quad (65)$$

Algebraic Supnick matrices can be defined analogously by replacing $+$ by \oplus and \leq by \preceq . *Inverse Supnick matrices* are obtained by reversing the inequality signs in (65).

Note that property (65) does not involve the diagonal entries c_{ii} (they need not be given at all). This is due to the fact that Supnick matrices were introduced by Supnick [120] in connection with polynomially solvable special cases of the travelling salesman problem. (A feasible tour never contains a loop.)

Burkard and Deineko (see [30]) observed that any Supnick matrix can be turned into a Monge matrix. More specifically, we have

Observation 9.4. *Let C be a symmetric $n \times n$ matrix. Then the following holds:*

- (i) *If C is a Monge matrix, then C is also a Supnick matrix.*
- (ii) *If C is a Supnick matrix, then the matrix \widehat{C} obtained from C by choosing the diagonal entries according to $\widehat{c}_{ii} := c_{i-1,i} + c_{i,i+1} - c_{i-1,i+1}$ for $i = 2, \dots, n-1$, $\widehat{c}_{11} := c_{21} + c_{12} - \widehat{c}_{22}$ and $\widehat{c}_{nn} := c_{n-1,n} - c_{n,n-1} - \widehat{c}_{n-1,n-1}$ is a Monge matrix.*

Obviously, inverse Supnick matrices and symmetric inverse Monge matrices are related in the same way. Statement (ii) becomes, however, false for algebraic Supnick matrices, e.g. bottleneck Supnick matrices can in general not be extended to bottleneck Monge matrices by choosing the diagonal entries in an appropriate way.

Observation 9.4 has two important consequences. First, it can be checked in $O(n^2)$ time whether or not a given $n \times n$ matrix is a Supnick matrix (first we determine the diagonal entries as described above and then we check whether the resulting matrix is a Monge matrix).

Secondly, it follows that the travelling salesman problem on Supnick matrices always has an optimal tour which is pyramidal (cf. Theorem 3.2). Supnick [120] showed the stronger result that if the cost matrix C is a Supnick matrix (symmetric Monge matrix), then the tour $\langle 1, 3, 5, \dots, n, \dots, 6, 4, 2 \rangle$ is optimal. This result has been extended to the algebraic (TSP) on algebraic Supnick matrices by Burkard and van der Veen [39].

Similarly, it follows that the (TSP) on inverse Supnick matrices is again pyramidal solvable. Supnick [120] proved that if the cost matrix C is an inverse Supnick matrix

(symmetric inverse Monge matrix), then an optimal tour is given by $\langle n, 2, n - 2, 4, n - 4, 6, \dots, 5, n - 3, 3, n - 1, 1 \rangle$. These results were rediscovered by Rubinshtein [109] and Michalski [95]. A simpler proof of the optimality of Supnick's permutations which makes use of Lemma 2.3 can be found in Rudolf and Woeginger [113].

Supnick's result for the (TSP) motivates to ask whether, given an $n \times n$ symmetric matrix C , there exists a permutation ϕ such that the matrix $C_{\phi,\phi}$ becomes a Supnick matrix. Note that in the case of the (TSP) rows and columns have to be permuted with the same permutation. This problem is solved in Deineko et al. [49] who give an algorithm which in $O(n^2 \log n)$ time either finds the desired permutation ϕ or proves that no such ϕ exists.

9.4. Incomplete Monge matrices

Supnick matrices can also be regarded as symmetric Monge matrices where the diagonal entries may remain unspecified. The following definition from Deineko et al. [49] extends this concept.

An $m \times n$ matrix C some of whose elements may remain unspecified is called an *incomplete Monge matrix* if whenever the entries c_{ij}, c_{rs}, c_{is} and c_{rj} , $1 \leq i \leq r \leq m$ and $1 \leq j \leq s \leq n$ are all specified, they satisfy

$$c_{ij} + c_{rs} \leq c_{rj} + c_{is}.$$

Unfortunately, it turns out that deciding whether, given an $m \times n$ matrix C , there exist permutations ϕ and ψ such that $C_{\phi,\psi}$ is an incomplete Monge matrix, is an NP-complete problem (see [49]).

9.5. Demidenko and Kalmanson matrices

Two other Monge-like types of matrices which have been investigated in connection with the (TSP) are symmetric *Demidenko* and symmetric *Kalmanson matrices*.

A symmetric $n \times n$ matrix C is called a symmetric *Demidenko matrix* if and only

$$c_{ij} + c_{rs} \leq c_{is} + c_{rj} \quad \text{for all } 1 \leq i < j < s < r \leq n.$$

and C is called a symmetric *Kalmanson matrix*, if we additionally require that

$$c_{ir} + c_{js} \leq c_{is} + c_{jr} \quad \text{for all } 1 \leq i < j < s < r \leq n. \quad (66)$$

Demidenko [51] showed that the (TSP) restricted to the class of symmetric Demidenko matrices is pyramidally solvable. For symmetric Kalmanson matrices this result can be strengthened to show that an optimal tour is always given by $\langle 1, 2, 3, \dots, n \rangle$ (cf. Kalmanson [80]).

Deineko et al. [50] give an $O(n^2 \log n)$ algorithm which, given an $n \times n$ matrix C , either finds a permutation ϕ such that matrix $C_{\phi,\phi}$ becomes a Kalmanson matrix or proves that no such permutation exists. The analogous problem for symmetric Demidenko matrices is still open. A partial result is obtained in Deineko et al. [48] where

it is shown that for the special case of $n \times n$ matrices C which arise as distance matrices of n points in the Euclidean plane, the above restricted recognition problem for permuted Demidenko matrices can be solved in $O(n^4)$ time.

9.6. The box inequalities

An $m \times n$ matrix C is said to satisfy the so-called *box inequalities* if for all $1 \leq i, r \leq m$ and $1 \leq j, s \leq n$

$$c_{ij} + c_{rs} \geq c_{is} + c_{rj} \quad \text{holds whenever } c_{ij} \geq \max\{c_{is}, c_{rj}, c_{rs}\}. \quad (67)$$

Similarly, C satisfies the *reverse box inequalities* if for all $1 \leq i, r \leq m$ and $1 \leq j, s \leq n$

$$c_{ij} + c_{rs} \leq c_{is} + c_{rj} \quad \text{holds whenever } c_{ij} \leq \min\{c_{is}, c_{rj}, c_{rs}\}. \quad (68)$$

How are these box inequalities related to Monge matrices resp. Monge sequences? First, observe that if $c_{ij} = \max\{c_{is}, c_{rj}, c_{rs}\}$ holds in (67), then we necessarily must have $c_{ij} + c_{rs} = c_{is} + c_{rj}$. Hence it is easy to construct (permuted) inverse Monge matrices which do not satisfy the box inequalities and (permuted) Monge matrices which do not satisfy the reverse box inequalities. What can be shown, however, rather easily is that any matrix which satisfies the box inequalities or the reverse box inequalities has a Monge sequence.

Applications of the box inequalities. Let $G = (V_1, V_2; E)$ be a weighted bipartite graph and let $C = (c_{ij})$ be its associated weight matrix. Suppose we are interested into finding a matching $M \subseteq E$ with maximal resp. minimal weight.

In Section 3 we have seen that this problem can be solved greedily if C has a Monge sequence and if $|V_1| = |V_2|$. The north-west corner rule, however, in general fails to find an optimal solution for the unbalanced case $|V_1| \neq |V_2|$ which arises quite often in practical applications.

It turns out that the box inequalities play an important role in deriving a class of greedily solvable unbalanced weighted bipartite matching problems. Consider the following simple greedy approach: Start with $M' := \emptyset$ and $E' := E$. As long as E' is not empty, pick an edge e in E' with largest (smallest) weight, add e to M' and delete e and all edges incident with e from E' .

It is well known (see e.g. [71]) that for arbitrary nonnegative real weights c_{ij} this approach yields a matching M' with at least half (at most half) the weight of the optimum matching. Blum et al. [24] identified the following special case where the greedy matching M' is optimal.

Theorem 9.5. *If the weight matrix C is nonnegative and satisfies the box inequalities, then the greedy algorithm, which always takes a maximum weight edge, determines a maximum weight matching.*

Similarly, it can be shown that if C is nonnegative and satisfies the reverse box inequalities, taking successively the edge with minimum weight leads to a minimum weight matching.

Matching problems where the weights satisfy the (reverse) box inequalities arise e.g. in connection with the shortest superstring problem which can be described as follows (cf. [24, 71, 123]): Given a set of strings, find the shortest string which contains each of the smaller strings as a contiguous substring. This problem is NP-hard, but plays an important role in molecular biology (DNA sequencing). In [24] two approximation algorithms are given, with performance ratios of 4 resp. 3, where in both algorithms a maximum weighted matching problem with weights satisfying the box inequalities has to be solved. Since the involved graphs can become quite large, a fast greedy algorithm is very helpful in practice.

9.7. The quasi-convex property

Let $G = (V_1, V_2; E)$ be a weighted bipartite graph with $|V_1| = m$ and $|V_2| = n$ and let $C = (c_{ij})$ denote its associated weighted adjacency matrix. The graph G is said to have the *quasi-convex property* if the nodes in V_1 and V_2 can be arranged on a circle \mathcal{C} such that the condition $c_{is} + c_{rj} \leq c_{ij} + c_{rs}$ holds whenever the vertices $i, r \in V_1$ and $j, s \in V_2$ occur in the order i, r, j, s in clockwise order around \mathcal{C} . Viewed geometrically, this condition simply requires that the sum of the weights of the diagonals (i, j) and (r, s) of the quadrilateral with vertices $i, r \in V_1$ and $j, s \in V_2$ is at least as large as the sum of the weights of the two sides (i, s) and (r, j) . (Note that this condition is nothing else than the second condition (66) required for Kalmanson matrices.)

In order to investigate the relationship between the quasi-convex property and the Monge property, consider the special case of the quasi-convex property where there exists an arrangement of the vertices along the circle \mathcal{C} such that the vertices in V_1 and V_2 are separated from each other. It is easy to see that in that case the matrix C can be permuted into an inverse Monge matrix by rearranging the rows and columns according to the order in which the vertices in V_1 and V_2 occur along \mathcal{C} . Consequently, the quasi-convex property is weaker than the (inverse) Monge property and by the observation made above also weaker than the Kalmanson property.

In Section 3 we already mentioned that the minimum cost matching problem becomes easier if the underlying weights satisfy the (inverse) Monge condition. Buss and Yianilos [41] show that this remains true for the weaker quasi-convex property. In this way, they generalize or improve upon results of Marcotte and Suri [94], Karp and Li [84] and Aggarwal et al. [4]. Specifically, Buss and Yianilos obtain an $O(n \log n)$ time algorithm for the quasi-convex balanced minimum cost matching problem where $n = m$ and even a linear time algorithm for the special case where the weights c_{ij} fulfill an additional condition, the so-called weak analyticity condition, and where a quasi-convex arrangement of the vertices is given. In some cases their results even apply to the unbalanced case where $n \neq m$.

10. Summary and concluding remarks

In this paper we gave a survey on the recently very active field of Monge properties and their role in combinatorial optimization and related areas. We mainly focused on the following three aspects:

- (i) Investigation of the fundamental combinatorial properties of Monge structures.
- (ii) Applications of Monge properties in combinatorial optimization and related fields.
- (iii) Recognition of Monge properties.

Furthermore, we made an effort to present the results already known from the literature in a unified way. In particular, we tried to stress the connections and relationships between the many different Monge properties known so far. Hopefully this will stimulate further research on Monge properties and Monge-like properties.

Lots of problems in connection with Monge properties remain, however, unsolved. Let us mention just a few open questions which we consider as particularly interesting ones.

(1) Since it turned out that many hard optimization problems become polynomial time solvable when restricted to Monge structures, the question on how this might be exploited for obtaining heuristics and approximation algorithms for general instances suggests itself. For example, we might ask the following: Given a cost matrix C which is not Monge and an instance I of a hard optimization problem P , find a Monge matrix C' such that the value of the optimal solution of the modified instance I' with cost matrix C' is as close as possible to the optimal solution of the original instance I . We are not aware of any work in this area.

(2) Investigate further classes of NP-hard optimization problems with respect to the question whether they become easier if the underlying cost structure has the Monge property. For example, the complexity of the max-cut problem and of the maximum clique problem on Monge matrices is not known. Concerning the notoriously hard quadratic assignment problem (QAP) some partial results have been obtained recently. Unlike for the problems treated in this survey, the presence of the Monge property alone does not turn the (QAP) into a polynomially solvable problem. In particular, the (QAP) remains NP-hard when the matrices A and B are both inverse Monge matrices and hence also when both A and B are Monge matrices (see [33, 32]). Further results concerning the (QAP) on Monge and Monge-like matrices are discussed in the forthcoming survey by Burkard et al. [32] and in the thesis of Çela [44].

(3) In Section 7 we gave a universal recognition algorithm for the class of permuted algebraic Monge matrices in strongly cancellative semigroups. Does there also exist a universal algorithm for weakly cancellative semigroups?

(4) Several interesting open questions are related to searching and selection problems in multi-dimensional Monge arrays. For example, the best known algorithm to identify all plane minima of a d -dimensional $n \times n \times \cdots \times n$ Monge array runs in $O(n \log^{d-2} n)$ time (cf. [5, 6], but the best known lower bound is $\Omega(dn)$). Since the algorithm of

Aggarwal and Park only exploits the total monotonicity condition, but not the stronger Monge property, it is conceivable, that at least for the case $d = 3$ an algorithm with running time $O(n \log n)$ exists. For further open problems in this area, see Aggarwal and Park [5–7].

(5) There is a deep connection between discrete and continuous Monge properties as can, e.g. be seen by the analogies between the discrete transportation problem and the continuous transportation problem treated in Section 9. These connections are not well investigated (cf. [17]), but will very likely lead to new and important results for both discrete and continuous problems.

(6) Matrices and arrays arising in practical problems often will not have the desired Monge property. This motivates to ask for the largest submatrix or subarray with this property. Questions of this type seem to be hard. A first result in this regard is due to Woeginger [134] showing that it is NP-hard to find the largest submatrix of a given $m \times n$ matrix which is Monge or permuted Monge, respectively. Of related interest is to find the largest submatrix which has a Monge sequence.

(7) Do there exist other Monge-like properties which have not been discussed in this paper? Are there any other fields of mathematics where Monge structures play a role?

Acknowledgements

We would like to thank James K. Park for many helpful remarks and for providing us with a copy of his thesis, and Vladimir Deineko for sharing with us his extensive knowledge of the Russian literature in the Monge area. Furthermore, we are grateful to Gerhard J. Woeginger for several stimulating discussions on the recognition of permuted Monge structures and to Alok Aggarwal, Ulrich Faigle and Gerhard J. Woeginger for feedback on a first version of this paper.

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, Applications and Complexity*, (World Scientific, Singapore, 1993) 1–22.
- [3] A. Aggarwal, A. Bar-Noy, S. Khuller, D. Kravets and B. Schieber, Efficient minimum cost matching using quadrangle inequality, in: *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science* (1992) 583–592.
- [4] A. Aggarwal, M.M. Klawe, S. Moran, P. Shor and R. Wilber, Geometric applications of a matrix-searching algorithm, *Algorithmica* 2 (1987) 195–208.
- [5] A. Aggarwal and J.K. Park, Notes on searching in multidimensional monotone arrays, in: *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, (1988) 497–512.
- [6] A. Aggarwal and J.K. Park, Sequential searching in multidimensional monotone arrays, Research Report RC 15128, IBM T.J. Watson Research Center, Yorktown Heights, NY, November 1989, portions of this paper appeared in [5].

- [7] A. Aggarwal and J.K. Park, Parallel searching in multidimensional monotone arrays, Research Report RC 14826, IBM T.J. Watson Research Center, Yorktown Heights, NY, November 1989, portions of this paper appeared in [5].
- [8] A. Aggarwal and J.K. Park, Improved algorithms for economic lot-size problems, Oper. Res. 41 (1993) 549–571.
- [9] A. Aggarwal and J.K. Park, More applications of Monge-array techniques to economic lot-size problems, unpublished manuscript, 1993.
- [10] A. Aggarwal, B. Schieber and T. Tokuyama, Finding a minimum weight K -link path in graphs with Monge property and applications, Discrete Comput. Geometry 12 (1994) 263–280.
- [11] P.K. Agarwal and S. Sen, Selection in monotone matrices and k th nearest neighbors, in: SWAT 94: Proceedings of the 4th Scandinavian Workshop on Algorithm Theory Lecture Notes in Computer Science 824, (Springer, Berlin, 1994) 13–24.
- [12] V.S. Aizenshtat and D.N. Kravchuk, Algorithms for finding extremum of a linear form on the set of all cycles in special cases, Dokl. Akad. Nauk BSSR 12 (1968) 401–404 (in Russian).
- [13] V.S. Aizenshtat and E.P. Maksimovich, Special cases of traveling salesman problems, Kibernetika 4 (1978) 80–83 (in Russian); Cybernetics 14 (1979) 565–569 (English translation).
- [14] 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.
- [15] E.J. Anderson and P. Nash, Linear Programming in Infinite-Dimensional Spaces: Theory and Applications, Wiley-Interscience Series in Discrete Mathematics and Optimization (Wiley, Chichester, 1987).
- [16] A. Apostolico, M.J. Atallah, L.L. Larmore and H.S. McFaddin, Efficient parallel algorithms for string editing and related problems, SIAM J. Comput. 19 (1990) 968–988.
- [17] M.L. Balinski and S.T. Rachev, On Monge–Kantorovich problems, preprint, SUNY at Stony Brook, Department of Applied Mathematics and Statistics, 1989.
- [18] E.R. Barnes and A.J. Hoffman, On transportation problems with upper bounds on leading rectangles, SIAM J. Algebraic Discrete Methods 6 (1985) 487–496.
- [19] W.W. Bein, P. Brucker and A.J. Hoffman, Series parallel composition of greedy linear programming problems, Math. Programming 62 (1993) 1–14.
- [20] W.W. Bein, P. Brucker and J.K. Park, Applications of an algebraic Monge property, unpublished manuscript, Presented at the 3rd Twente Workshop on Graphs and Combinatorial Optimization, Enschede, The Netherlands, June 1993.
- [21] 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.
- [22] W.W. Bein, P. Brucker and A. Tamir, Minimum cost flow algorithms for series-parallel networks, Discrete Appl. Math. 10 (1985) 117–124.
- [23] W. Bein and P.K. Pathak, A characterization of the Monge property, Technical Report CS90-10, Department of Computer Science, University of New Mexico, Albuquerque, New Mexico, September 1990.
- [24] A. Blum, T. Jiang, M. Li, J. Tromp and M. Yannakakis, Linear approximation of shortest superstrings, in: Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing, (1991) 328–336.
- [25] A. Borchers and P. Gupta, Extending the quadrangle inequality to speed-up dynamic programming, Inform. Process. Lett. 49 (1994) 287–290.
- [26] V.Y. Burdyuk and V.N. Trofimov, Generalization of the results of Gilmore and Gomory on the solution of the traveling salesman problem, Izv. Akad. Nauk SSSR, Tech. Kibernet. 3 (1976) 16–22 (in Russian); Engi. Cybernet. 14 (1976) 12–18 (English translation).
- [27] R.E. Burkard, Remarks on some scheduling problems with algebraic objective function, Methods Oper. Res. 32 (1978) 63–77.
- [28] R.E. Burkard, A general Hungarian method for the algebraic transportation problem, Discrete Math. 22 (1978) 219–232.
- [29] R.E. Burkard, Assignment problems: recent solution methods and applications, in: Proceedings of the 12th IFIP Conference: System Modelling and Optimization, Budapest, Hungary, 1985, Springer Lecture Notes in Control and Information Science 84 (Springer, Berlin, Heidelberg, New York, 1986) 153–169.
- [30] R.E. Burkard, Special cases of travelling salesman problems and heuristics, Acta Math. Appl. Sinica 6 (1990) 273–288.

- [31] R.E. Burkard, On the role of bottleneck Monge matrices in combinatorial optimization, *Oper. Res. Lett.* 17 (1995) 53–56.
- [32] R.E. Burkard, E. Çela, V.M. Demidenko, N.N. Metelski and G.J. Woeginger, Easy and hard cases of the quadratic assignment problem: a survey, manuscript, Institute of Mathematics, University of Technology Graz, Austria, 1995.
- [33] R.E. Burkard, E. Çela, G. Rote and G.J. Woeginger, The quadratic assignment problem with an Anti-Monge and a Toeplitz matrix: easy and hard cases, SFB-Report No. 34, SFB “Optimierung und Kontrolle”, Institute of Mathematics, University of Technology, Graz, Austria, June 1995, Mathematical Programming, to appear.
- [34] R.E. Burkard and V.G. Deineko, Polynomially solvable cases of the traveling salesman problem and a new exponential neighborhood, *Computing* 54 (1995) 191–211.
- [35] R.E. Burkard, V.G. Deineko, R. van Dal, J.A.A. van der Veen and G.J. Woeginger, Well-solvable special cases of the TSP: A survey, Institute of Mathematics, University of Technology, Graz, Austria, 1995.
- [36] R.E. Burkard, V.G. Deineko and G.J. Woeginger, The traveling salesman problem on permuted Monge matrices, SFB-Report No. 31, SFB “Optimierung und Kontrolle”, Institute of Mathematics, University of Technology, Graz, Austria, 1995, submitted.
- [37] R.E. Burkard, R. Rudolf and G.J. Woeginger, Axial assignment problems with decomposable cost coefficients, Report 238-92, Institute of Mathematics, University of Technology, Graz, Austria, October 1992, *Discrete Appl. Math.*, to appear.
- [38] R.E. Burkard and W. Sandholzer, Efficiently solvable special cases of bottleneck travelling salesman problems, *Discrete Appl. Math.* 32 (1991) 61–76.
- [39] R.E. Burkard and J.A.A. van der Veen, Universal conditions for algebraic travelling salesman problems to be efficiently solvable, *Optimization* 22 (1991) 787–814.
- [40] R.E. Burkard and U. Zimmermann, Weakly admissible transformations for solving algebraic assignment and transportation problems, *Math. Programming Study* 12 (1980) 1–18.
- [41] S.R. Buss and P.N. Yianilos, Linear and $O(n \log n)$ time minimum-cost matching algorithms for quasi-convex tours, in: *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms* (1994) 65–76.
- [42] S. Cambanis, G. Simons and W. Stout, Inequalities for $E_k(X, Y)$ when the marginals are fixed, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 36 (1976) 285–294.
- [43] K. Cechlárová and P. Szabó, On the Monge property of matrices, *Discrete Math.* 81 (1990) 123–128.
- [44] E. Çela, The quadratic assignment problem: special cases and relatives, Ph.D. Thesis, Institute of Mathematics, University of Technology, Graz, Austria, October 1995.
- [45] L. Chen and Y. Yesha, Efficient parallel algorithms for bipartite permutation graphs, *Networks* 22 (1993) 29–39.
- [46] V.G. Deineko, Applying dynamic programming to solving a special multi-traveling salesmen problem, *Issledovanie operazi i ASU* 14 Kiev, (1979) 47–50, (in Russian).
- [47] V.G. Deineko and V.L. Filonenko, On the reconstruction of specially structured matrices, *Aktualnyje Problemy EVM i programmirovaniye*, Dnepropetrovsk, DGU, 1979 (in Russian).
- [48] V.G. Deineko, R. Rudolf, J.A.A. van der Veen and G.J. Woeginger, Three easy special cases of the Euclidean travelling salesman problem, SFB-Report No. 17, SFB “Optimierung und Kontrolle”, Institute of Mathematics, University of Technology, Graz, Austria, January 1995, submitted for publication.
- [49] V.G. Deineko, R. Rudolf and G.J. Woeginger, The recognition of permuted Supnik and incomplete Monge matrices, SFB-Report No. 5, SFB “Optimierung und Kontrolle”, Institute of Mathematics, University of Technology, Graz, Austria, August 1994, *Acta Inform.*, to appear.
- [50] V.G. Deineko, R. Rudolf and G.J. Woeginger, Sometimes travelling is easy: the master tour problem, SFB-Report No. 13, SFB “Optimierung und Kontrolle”, Institute of Mathematics, University of Technology, Graz, Austria, December 1994, submitted for publication. An extended abstract appeared in *Lecture Notes in Computer Science* 979 (Springer, Berlin, 1995) 128–141.
- [51] V.M. Demidenko, A special case of travelling salesman problems, *Izv. Akad. Nauk. BSSR, Ser. fiz.-mat. nauk* 5 (1976) 28–32 (in Russian).
- [52] U. Derigs, O. Goecke and R. Schrader, Bisimplicial edges, Gaussian elimination and matchings in bipartite graphs, in: Pape, U. ed., *Graph-theoretic Concepts in Computer Science*, (Trauner Verlag, Linz, 1984) 79–87.

- [53] U. Derigs, O. Goecke and R. Schrader, Monge sequences and a simple assignment algorithm, *Discrete Appl. Math.* 15 (1986) 241–248.
- [54] B.L. Dietrich, Monge sequences, antimatroids and the transportation problem with forbidden arcs, *Linear Algebra Appl.* 139 (1990) 133–145.
- [55] H.A. Eiselt and R.E. Burkard, Reshipments and overshipments in transportation problems with minimax objective, *OR Spektrum* 13 (1991) 133–140.
- [56] D. Eppstein, Sequence comparison with mixed convex and concave costs, *J. Algorithms* 11 (1990) 85–101.
- [57] R.E. Erickson, C.L. Monma and A.F. Veinott Jr., Send-and-split method for minimum-concave-cost network flows, *Math. Oper. Res.* 12 (1987) 634–664.
- [58] U. Faigle, Some recent results in the analysis of greedy algorithms for assignment problems, *OR Spektrum* 15 (1994) 181–188.
- [59] U. Faigle and W. Kern, Submodular linear programs on forests, Memorandum No. 1163, University of Twente, September 1993, *Math. Programming*, to appear.
- [60] A. Federgruen and M. Tzur, A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(n \log n)$ or $O(n)$ time, *Management Sci.* 37 (1991) 909–925.
- [61] D. Fortin and R. Rudolf, Weak algebraic Monge arrays, SFB-Report No. 33, SFB “Optimierung und Kontrolle”, Institute of Mathematics, University of Technology, Graz, Austria, June 1995, submitted for publication.
- [62] M. Fréchet, Sur les tableaux de corrélation dont les marges sont données, *Annales de l'Université de Lyon, Section A* 14 (1951) 53–77.
- [63] M. Fréchet, Sur les tableaux de corrélation dont les marges sont données, *Annales de l'Université de Lyon, Section A* 20 (1957) 13–31.
- [64] N.E. Gaikov, On the minimization of a linear form on cycles, *Vestsi Akad. Navuk BSSR Ser. Fiz.-Mat. Navuk* 4 (1980) 128 (in Russian).
- [65] Z. Galil and K. Park, A linear-time algorithm for concave one-dimensional dynamic programming, *Inform. Process. Lett.* 33 (1990) 309–311.
- [66] P.C. Gilmore, E.L. Lawler and D.B. Shmoys, Well-solved special cases, in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (eds.), *The Traveling Salesman Problem*, Chap. 4 (Wiley, Chichester, 1985) 87–143.
- [67] E. Girschich, M.M. Kovalev and A.A. Zaporozhets, Subcones of submodular functions (subclasses of Monge matrices), Preprint No. 29, Fakultät für Mathematik, Otto-von-Guericke Universität Magdeburg, Germany, September 1994.
- [68] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, (Academic Press, New York, 1980).
- [69] M.C. Golumbic and C.F. Goss, Perfect elimination and chordal bipartite graphs, *J. Graph Theory* 2 (1978) 155–163.
- [70] S.C. Graves and J.B. Orlin, A minimum concave-cost dynamic network flow problem with an application to lot-sizing, *Networks* 15 (1985) 59–71.
- [71] D. Gusfield, Design (with analysis) of efficient algorithms, in: E.G. Coffman et al. eds., *Handbooks in OR & MS*, Vol. 3, Computing, Chap. 8 North-Holland, (Elsevier, 1992) 375–453.
- [72] G.H. Hardy, J.E. Littlewood and G. Pólya, *Inequalities*, (Cambridge Univ. Press, Cambridge, 1952).
- [73] D.S. Hirschberg and L.L. Larmore, The least weight subsequence problem, *SIAM J. Comput.* 16 (1987) 628–638.
- [74] W. Hoeffding, Maßstabsinvariante Korrelationstheorie, *Schriften des Mathematischen Instituts und des Instituts für Angewandte Mathematik der Universität Berlin* 5 (1940) 179–233.
- [75] A.J. Hoffman, On simple linear programming problems, in: V. Klee ed. *Proceedings of Symposia in Pure Mathematics, Convexity*, Vol. VII (AMS, Providence, RI, 1963) 317–327.
- [76] A.J. Hoffman, On greedy algorithms that succeed, in: Anderson, ed., *Surveys in combinatorics*, London Mathematical Society Lecture Notes Series 103 1. (Cambridge Univ. Press, Cambridge, 1985) 97–112.
- [77] A.J. Hoffman, On greedy algorithms for series parallel graphs, *Math. Programming* 40 (1988) 197–204.
- [78] A.J. Hoffman and A.F. Veinott, Staircase transportation problems with superadditive rewards and cumulative capacities, *Math. Programming* 62 (1993) 199–213.
- [79] S.M. Johnson, Optimal two and three-stage production schedules with setup times included, *Naval Res. Logistics Quart.* 1 (1954) 61–68.

- [80] K. Kalmanson, Edgeconvex circuits and the traveling salesman problem, *Can. J. Math.* 27 (1975) 1000–1010.
- [81] L.V. Kantorovich, On the transfer of masses, *Doklady Akad. Nauk. USSR* 37 (1942) 227–229.
- [82] L.V. Kantorovich, On a problem of Monge, *Uspekhi Mat. Nauk.* 3 (1948) 225–226 (in Russian).
- [83] S. Karlin, *Total Positivity*, Vol. 1, (Stanford University Press, Stanford, CA, 1968).
- [84] R.M. Karp and S-Y.R. Li, Two special cases of the assignment problem, *Discrete Math.* 13 (1975) 129–142.
- [85] M.M. Klawe and D.J. Kleitman, An almost linear time algorithm for generalized matrix searching, *SIAM J. Discrete Math.* 3 (1990) 81–97.
- [86] B. Klinz, R. Rudolf and G.J. Woeginger, Permuting matrices to avoid forbidden submatrices, *Discrete Appl. Math.* 60 (1995) 223–248.
- [87] B. Klinz, R. Rudolf and G.J. Woeginger, On the recognition of permuted bottleneck Monge matrices, *Discrete Appl. Math.* 63 (1995) 43–74.
- [88] B. Korte, L. Lovász and R. Schrader, *Greedoids*, (Springer, Berlin, 1991).
- [89] D. Kravets and J.K. Park, Selection and sorting in totally monotone matrices, *Math. Systems Theory* 24 (1991) 201–220.
- [90] L.L. Larmore and B. Schieber, On-line dynamic programming with applications to the prediction of RNA secondary structure, *J. Algorithms* 12 (1991) 490–515.
- [91] G.G. Lorentz, An inequality for rearrangements, *Amer. Math. Monthly* 60 (1953) 176–179.
- [92] L. Lovász, Submodular functions and convexity, in: A. Bachem, M. Grötschel, B. Korte, eds., *Mathematical Programming: The State of the Art*, Bonn 1982, (Springer, Berlin, 1983) 235–257.
- [93] Y. Mansour, J.K. Park, B. Schieber and S. Sen, Improved selection in totally monotone arrays, *Int. J. Comput. Geometry Appl.* 3 (1993) 115–132.
- [94] O. Marcotte and S. Suri, Fast matching algorithms for points on a polygon, *SIAM J. Comput.* 20 (1991) 405–422.
- [95] M. Michalski, On a class of polynomially solvable travelling salesman problems, *Zastosowania Matematyki* 19 (1987) 531–539.
- [96] G. Monge, Mémoire sur la théorie des déblais et des remblais, in: *Histoire de l'Académie Royale des Sciences, Année M. DCCLXXXI*, avec les Mémoires de Mathématique et de Physique, pour la même Année, Tirés des Registres de cette Académie, Paris, (1781) 666–704.
- [97] C.L. Monma and A.H.G. Rinnooy Kan, A concise survey of efficiently solvable special cases of the permutation flow shop-problem, *RAIRO Recherche opérationnelle* 17, (1983) 105–119.
- [98] I. Olkin and S.T. Rachev, Distributions with given marginals subject to certain constraints, Technical Report No. 270, Department of Statistics, Stanford University, Stanford, CA, 1990 (appeared also as Technical Report No. 129, Department of Statistics and Applied Probability, University of California, Santa Barbara, 1990).
- [99] I. Olkin and S.T. Rachev, Mass transportation problems with capacity constraints, Report 9154/A, Econometric Institute, Erasmus University, Rotterdam, The Netherlands, 1991.
- [100] J.B. Orlin, A faster strongly polynomial minimum cost flow algorithm, in: *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing*, (1988) 377–387.
- [101] J. O'Rourke, A. Aggarwal, S. Maddila and M. Baldwin, An optimal algorithm for finding minimal enclosing triangles, *J. Algorithms* 7 (1986) 258–268.
- [102] J.K. Park, The Monge array: an abstraction and its applications, Ph.D. Thesis, MIT, Massachusetts, 1991.
- [103] J.K. Park, A special case of the n -vertex traveling salesman problem that can be solved in $O(n)$ time, *Inform. Process. Lett.* 40 (1991) 247–254.
- [104] U. Pferschy, R. Rudolf and G.J. Woeginger, Monge matrices make maximization manageable, *Oper. Res. Lett.* 16 (1994) 245–254.
- [105] M. Queyranne, F. Spieksma and F. Tardella, A general class of greedily solvable linear programs, in: *Proceedings of the Third IPCO Conference* (1993) 385–399.
- [106] S.T. Rachev, The Monge-Kantorovich mass transference problem and its stochastic applications, *Theory Probab. Appl.* 29 (1984) 647–676, (translated from *Teoriya Veroyatnostei i ee Primeneniya*).
- [107] S.T. Rachev, *Probability Metrics and the Stability of Stochastic Models*, (Wiley, Chichester, 1991).
- [108] S.T. Rachev and L. Rüschendorf, Solution of some transportation problems with relaxed or additional constraints, *SIAM J. Control Optim.* 32 (1994) 673–689.

- [109] M.I. Rubinshtein, On the symmetric traveling salesman problem, *Automat. Remote Control* 32 (1971) 1453–1460 (translated from *Avtomatika i Telemekhanika*).
- [110] R. Rudolf, Monge properties and their recognition, Ph.D. Thesis, Technical University Graz, 1993.
- [111] R. Rudolf, Recognition of d -dimensional Monge arrays, *Discrete Appl. Math.* 52 (1994) 71–82.
- [112] R. Rudolf, On Monge sequences in d -dimensional arrays, Report 247-93, Institute of Mathematics, University of Technology, Graz, Austria, April 1993, submitted for publication.
- [113] R. Rudolf and G.J. Woeginger, The cone of Monge matrices: extremal rays and applications, *ZOR Methods Models Oper. Res.* 42 (1995) 161–168.
- [114] V.I. Sarvanov, On quadratic choice problems, Ph.D. Thesis, Inst. Mat. AN BSSR, 1978, Minsk. (in Russian).
- [115] B. Schieber, Computing a minimum-weight k -link path in graphs with the concave Monge property, in: Proceedings of the Sixth ACM-SIAM Symposium on Algorithms, (1995) 405–411.
- [116] R. Shamir, A fast algorithm for constructing Monge sequences in transportation problems with forbidden arcs, *Discrete Math.* 114 (1993) 435–444.
- [117] R. Shamir and B.L. Dietrich, Characterization and algorithms for greedily solvable transportation problems, in: Proceedings of the First ACM-SIAM Symposium of Discrete Algorithms, (1990) 358–366.
- [118] J. Spinrad, A. Brandstädt and L. Stewart, Bipartite permutation graphs, *Discrete Appl. Math.* 18 (1987) 279–292.
- [119] J. Spinrad, Nonredundant 1s in Γ -free matrices, *SIAM J. Discrete Math.* 8 (1995) 251–257.
- [120] F. Supnick, Extreme hamiltonian lines, *Ann. Math.* 66 (1957) 179–201.
- [121] W. Szwarc, An instant solution of the $2 \times n$ bottleneck transportation problem, *Oper. Res. Lett.* 14 (1993) 261–264.
- [122] A.H. Tchen, Inequalities for distributions with given marginals, *The Ann. Probab.* 8 (1980) 814–827.
- [123] S.-H. Teng and F. Yao, Approximating shortest superstrings, in: Proceedings of the 34-th Annual IEEE Symposium on Foundations of Computer Science, (1993) 158–165.
- [124] D.M. Topkis, Minimizing a submodular function on a lattice, *Oper. Res.* 26 (1978) 305–321.
- [125] A. Tucker, A structure theorem for the consecutive ones property, *J. Combin. Theory* 12(B) (1972) 153–162.
- [126] R. Varadarajan, An optimal algorithm for $2 \times n$ bottleneck transportation problems, *Oper. Res. Lett.* 10 (1991) 525–529.
- [127] J.A.A. van der Veen, Solvable cases of the traveling salesman problem with various objective functions, Ph.D. Thesis, University Groningen, The Netherlands, 1992.
- [128] J.A.A. van der Veen, An $O(n)$ algorithm to solve the bottleneck traveling salesman problem restricted to ordered product matrices, *Discrete Appl. Math.* 47 (1993) 57–75.
- [129] A.F. Veinott, Jr., Unpublished class notes, Program in Operations Research, Stanford University, Stanford, CA, 1963.
- [130] A. Wagelmans, S. van Hoesel and A. Kolen, Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case, *Oper. Res.* 40 (1992) S145–156.
- [131] H.M. Wagner and T.M. Whitin, Dynamic version of the economic lot size model, *Management Sci.* 5 (1958) 89–96.
- [132] M.S. Waterman and T.F. Smith, Rapid dynamic programming algorithms for RNA secondary structure, *Advances Appl. Math.* 7 (1986) 455–464.
- [133] R. Wilber, The concave least-weight subsequence problem revisited, *J. Algorithms* 9 (1988) 418–425.
- [134] G.J. Woeginger, Personal communication, 1994.
- [135] F.F. Yao, Efficient dynamic programming using quadrangle inequalities, in: Proceedings of the 12-th Annual ACM Symposium on Theory of Computing, (1980) 429–435.
- [136] F.F. Yao, Speed-up in dynamic programming, *SIAM J. Algebraic Discrete Methods* 3 (1982) 532–540.
- [137] W.J. Zangwill, A deterministic multi-period production scheduling model with backlogging, *Management Sci.* 13 (1966) 105–119.
- [138] U. Zimmermann, Linear and combinatorial optimization in ordered algebraic structures, *Ann. Discrete Math.* 10; (North-Holland, Amsterdam, 1981).