V. G. DEINEKO
J. A. VAN DER VEEN
R. RUDOLF
G. J. WOEGINGER

Three easy special cases of the euclidean travelling salesman problem


<http://www.numdam.org/item?id=RO_1997__31_4_343_0>


L’accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l’accord avec les conditions générales d’utilisation (http://www.numdam.org/legal.php). Toute utilisation commerciale ou impression systématique est constitutive d’une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.
THREE EASY SPECIAL CASES OF THE EUCLIDEAN TRAVELLING SALESMAN PROBLEM (*)

by V. G. Deïneko (1) (2), J. A. van der Veen (3), R. Rudolf (2) and G. J. Woeginger (2)

Communicated by Philippe ChrÉtiennE

Abstract. — It is known that in case the distance matrix in the Travelling Salesman Problem (TSP) fulfills certain combinatorial conditions (e.g. the Demidenko conditions, the Kalmanson conditions or the Supnick conditions) then the TSP is solvable in polynomial time. This paper deals with the problem of recognizing Euclidean instances of the TSP for which there is a renumbering of the cities such that the corresponding renumbered distance matrix fulfills the Demidenko (Kalmanson, Supnick) conditions. We provide polynomial time recognition algorithms for all three cases.

Keywords: Travelling salesman problem, Kalmanson condition, Demidenko condition, Supnick condition, Combinatorial optimization, Geometry, Polynomial algorithms.

1. INTRODUCTION

The travelling salesman problem (RSP) is defined as follows. Given an \( n \times n \) distance matrix \( C = (c_{ij}) \) find a permutation \( \pi \in S_n \) that minimizes

\[ \sum_{i=1}^{n} c_{i\pi(i)} \]

(*) Manuscript received March 1995.
This research has been supported by Spezialforschungsbereich F 003 “Optimierung und Kontrolle”, Projektbereich Diskrete Optimierung.
(1) TU Graz, institut für Mathematik B, Steyrergasse 30, A-8010 Graz, Austria.
(2) On leave from Department of Applied Mathematics, Dnepropetrovsk University, Gagarin Av. 72, 320625 Dnepropetrovsk, Ukraine.
(3) Department of Quantitive Methods, Nijenrode, The Netherlands School of Business, Straatweg 25, 3621 BG Breukelen, The Netherlands.

Recherche opérationnelle/Operations Research, 0399-0559/97/04/$ 7.00 © AFCET-Gauthier-Villars
the sum $\sum_{i=1}^{n-1} c_{\pi(i) \pi(i+1)} + c_{\pi(n) \pi(1)}$ (the salesman must visit the cities 1 to $n$ in arbitrary order and wants to minimize the total travel length). This problem is one of the fundamental problems in combinatorial optimization and known to be NP hard. For more information, the reader is referred to the book by Lawler, Lenstra, Rinnooy Kan and Shmoys [8].

Several special cases of the TSP are solvable in polynomial time, due to special combinatorial structures in the distance matrix. A large class of such easy special cases is related to the concept of pyramidal tours, i.e. permutations $\pi \in S_n$ with $\pi = (1, i_1, i_2, \ldots, i_r, n, j_1, \ldots, j_{n-r-2})$ where $i_1 < i_2 < \ldots < i_r$ and $j_1 > \ldots > j_{n-r-2}$ hold (for permutations, we use the notation $\pi = (x_1, x_2, \ldots, x_n)$ for “$\pi(i) = x_i$ for $1 \leq i \leq n$”). Although the number of pyramidal tours on $n$ cities is exponential in $n$, a minimum cost pyramidal tour can be determined in $O(n^2)$ time by a dynamic programming approach (cf. Gilmore, Lawler and Shmoys [5]). For several classes of specially structured matrices it is known that these matrices always possess an optimal TSP tour which is pyramidal. Among these classes are the class $\mathbb{D}$ of Demidenko matrices, the class $\mathbb{K}$ of Kalmanson matrices and the class $\mathbb{S}$ of Supnick matrices. A symmetric $n \times n$ matrix $C$ is a Demidenko matrix ($C \in \mathbb{D}$) if

$$c_{ij} + c_{kl} \leq c_{ik} + c_{jl} \quad \text{for } 1 \leq i < j < k < l \leq n.$$  \hfill (1)

A symmetric matrix $C$ is a Kalmanson matrix ($C \in \mathbb{K}$), if it fulfills condition (1) and additionally

$$c_{il} + c_{jk} \leq c_{ik} + c_{jl} \quad \text{for } 1 \leq i < k < l \leq n.$$  \hfill (2)

A symmetric $n \times n$ matrix $C$ is a Supnik matrix ($C \in \mathbb{S}$) if

$$c_{ir} + c_{js} \leq c_{is} + c_{jr} \quad \text{for } 1 \leq i < j < n, \quad 1 \leq r < s \leq n,$$

$$\{i, j\} \cap \{r, s\} = \emptyset.$$  \hfill (3)

In a famous paper in 1976, Demidenko [3] proved that for the TSP with Demidenko distance matrices there always exists an optimal tour that is pyramidal. Consequently, the TSP with Demidenko distance matrices is efficiently solvable. Since $\mathbb{K} \subseteq \mathbb{D}$, this result immediately carries over to Kalmanson matrices. However, here an even stronger statement holds: For symmetric Kalmanson distance matrices, the (pyramidal) identity permutation
(1, 2, 3, ..., n) constitutes a shortest TSP tour (cf. Kalmanson [7]). Finally, for Supnick matrices the pyramidal permutation (1, 3, 5, 7, ..., 8, 6, 4, 2), i.e. first the odd cities in increasing order and then the even cities in decreasing order, yields an optimal tour (cf. Supnick [12]).

Another important special case of the TSP is the Euclidean TSP: here the cities are points in the two-dimensional plane and their distances are measured according to the Euclidean metric. It is easy to see that in this case, the shortest TSP tour does not intersect itself (cf. Flood [4]) and hence, geometry makes the problem somewhat easier. Nevertheless, this special case is still NP-hard (see e.g. Papadimitriou [6] or chapter 3 in the TSP book [8]).

The subject of this paper is to identify easy instances of the Euclidean TSP based on the concept of Demidenko (Kalmanson, Supnick) matrices: trivially, the length of the optimum TSP tour does not depend on the original numbering of the cities. However for some of the numberings, the distance matrix may fulfill the Demidenko (Kalmanson, Supnick) conditions whereas for other numberings it does not. Hence, the problem arises of finding numberings of the cities such that the resulting matrix fulfills the Demidenko (Kalmanson, Supnick) conditions. The corresponding algorithmic problem is called “recognition of permuted Euclidean Demidenko (Kalmanson, Supnick) matrices”. In this paper, we will derive the following results.

(a) Permuted n x n Euclidean Demidenko matrices can be recognized in O(n^4) time.

(b) Permuted n x n Euclidean Kalmanson matrices can be recognized in O(n^2) time.

(c) Permuted n x n Euclidean Supnick matrices are trivial to recognize: with a small number of exceptions only point sets in one-dimensional subspaces have Supnick distance matrices.

Our methods strongly exploit geometric structures in the problems like convex subsets and orderings along convex hulls, points lying on the branch of certain hyperbolas, intersection points of certain related hyperbolas and so on.

**Organization of the paper.** Sections 2 and 3 summarize elementary results and definitions for Kalmanson and Demino matrixes: Section 2 deals with combinatorial preliminaries, Section 3 with geometric preliminaries. The recognition problem of permuted Euclidean Kalmanson matrices is treated in Section 4 and permuted Euclidean Demidenko matrices are treated in Section 5. Section 6 gives a full characterization of Euclidean Supnick matrices. Finally, Section 7 closes with the discussion.
2. COMBINATORIAL PRELIMINARIES AND DEFINITIONS

In this section, several basic definitions for permutations and matrices are given and elementary properties of Demidenko, Kalmanson and Supnick matrices are summarized.

For an \( n \times n \) matrix \( C \), denote by \( I = \{1, \ldots, n\} \) the set of rows (columns). A row \( i \) precedes a row \( j \) in \( C \) (\( i < j \) for short), if row \( i \) occurs before row \( j \) in \( C \). For two sets \( K_1 \) and \( K_2 \) of rows, we write \( K_1 < K_2 \) iff \( k_1 < k_2 \) for all \( k_1 \in K_1 \) and \( k_2 \in K_2 \).

For \( V = \{v_1, v_2, \ldots, v_r\} \) a subset of \( I \), we denote by \( C[V] \) the \( r \times r \) submatrix of \( C \) which is obtained by deleting all rows and columns not contained in \( V \).

The identity permutation is denoted by \( e \), i.e. \( e(i) = i \) for all \( i \in I \). For a permutation \( \phi \), the permutation \( \phi^- \) defined by \( \phi^- (i) = \phi(n - i + 1) \) is called the reverse permutation of \( \phi \). Permutation \( \phi \) is called a cyclic shift or a rotation if there exists a \( k \in I \) such that \( \phi = (k, k+1, \ldots, n, 1, \ldots, k-1) \).

By \( C_\phi \) we denote the matrix which is obtained from matrix \( C \) by permuting its rows and columns according to \( \phi \), i.e. \( C_\phi = (c_{\phi(i), \phi(j)}) \). A permutation \( \phi \) is called a Demidenko (Kalmanson, Supnick) permutation for some matrix \( C \) iff \( C_\phi \) is a Demidenko (Kalmanson, Supnick) matrix.

For a partition \( X = \{X_1, \ldots, X_x\} \) of \( I \) into \( x \) subsets, the set \( S_{TR}(X_1, \ldots, X_x) \) contains all permutations \( \phi \) that fulfill \( \phi(x_1) < \phi(x_j) \) for all \( x_i \in X_i \) and \( x_j \in X_j \) with \( 1 \leq i < j \leq x \). \( S_{TR}(X_1, \ldots, X_x) \) is called the set of permutations induced by the sequence of stripes \( X_1, \ldots, X_x \).

For a partition \( X = \{X_1, \ldots, X_x\} \) of \( I \) into \( x \) subsets, the set \( S_{TR}(X_1, \ldots, X_x) \) contains all permutations \( \phi \) that fulfill \( \phi(x_1) < \phi(x_j) \) for all \( x_i \in X_i \) and \( x_j \in X_j \) with \( 1 \leq i < j \leq x \). \( S_{TR}(X_1, \ldots, X_x) \) is called the set of permutations induced by the sequence of stripes \( X_1, \ldots, X_x \).

Readers that are familiar with the concept of PQ-trees (Booth and Lueker [1]) may observe that the set \( S_{TR}(X_1, \ldots, X_x) \) can be represented by a PQ-tree of height two: the root is a Q-node with \( x \) sons. All sons of the root are P-nodes, where the \( i \)-th son has the elements in \( X_i \) as children.

**Proposition 2.1: (Booth and Lueker [1])** For two partitions \( \{X_1, \ldots, X_x\} \) and \( \{Y_1, \ldots, Y_y\} \) of \( I \), the set \( S_{TR}(X_1, \ldots, X_x) \cap S_{TR}(Y_1, \ldots, Y_y) \) either equals \( S_{TR}(Z_1, \ldots, Z_z) \) for an appropriate partition \( Z = \{Z_1, \ldots, Z_z\} \) of \( I \) or it is empty. The partition \( Z \) can be computed in \( O(|I|) \) time.

**Observation 2.2:** Let \( D \in \mathbb{D}, K \in \mathbb{K} \) and \( S \in \mathbb{S} \). Then \( D_{e^-} \in \mathbb{D}, K_{e^-} \in \mathbb{K} \) and \( S_{e^-} \in \mathbb{S} \) holds, and for any set \( J \subseteq I \), \( D[J] \in \mathbb{D}, K[J] \in \mathbb{K} \), and \( S[J] \in \mathbb{S} \). Moreover, for any cyclic shift \( \sigma \), \( K_{\sigma} \in \mathbb{K} \).
In other words, reversing a matrix does not destroy the combinatorial structures we are interested in and cyclically renumbering the rows and columns transforms Kalmanson matrices into Kalmanson matrices. For two rows (columns) $i$ and $j$ of $C$, define

$$M(i, j) = \{k \in I \setminus \{i, j\} | c_{ik} - c_{jk} = \min_{l \neq i, j} \{c_{il} - c_{jl}\}\}.$$ 

**Lemma 2.3:** Let $C$ be a symmetric $n \times n$ Kalmanson matrix with $n \geq 4$. Let $i$ and $j$ be two rows of $C$ with $i < j$, $K = M(i, j) \cup \{i\}$ and $K' = I \setminus K$. Then there exists a cyclic shift $\phi$ such that $C\phi \in \mathcal{K}$ and $K < K'$ in $C\phi$.

**Proof:** By definition $i \in K$ and $j \in K'$. Consider some $k \in M(i, j)$. Then $c_{ik} - c_{jk} = c_{il} - c_{jl}$ for all $l \in K \setminus \{i\}$ and $c_{ik} - c_{jk} < c_{il} - c_{jl}$ for all $l \in K' \setminus \{j\}$. Let $I' = I \setminus \{i, j, k\}$. Distinguish the following three cases on the relative position of $i$, $j$ and $k$: (i) $k < i < j$. The condition (2) implies $p \in K$ for all $p \in I'$ with $k < p < i$. (ii) $i < k < j$. By condition (1) $p \in K$ for all $p \in I'$ with $i < p < k$. (iii) $i < j < k$. Since $C \in \mathcal{K}$, $p \in K$ for all $p \in I'$ with $k < p$ or $p < i$.

Summarizing, there exist two elements $r$ and $s$ such that either $K = \{r, \ldots, i, \ldots, s\}$ or $K' = \{s+1, \ldots, j, \ldots, r-1\}$. By Observation 2.2 every cyclic shift of $C$ yields again a Kalmanson matrix. Choosing $\phi = \langle r, \ldots, s, \ldots, n, 1, \ldots, r-1 \rangle$ or $\phi = \langle r, \ldots, n, 1, \ldots, s, s+1, \ldots, r-1 \rangle$ completes the argument. £

Sometimes it is useful to use other, equivalent characterizations of the specially structured matrices. One such characterization of $\mathcal{D}$ was given in [5]:

**Observation 2.4:** ([5]) A symmetric $n \times n$ matrix $C$ is a Demidenko matrix iff

$$c_{ij} + c_{j+1,l} \leq c_{i,j+1} + c_{j,l} \quad \text{for all } 1 \leq i < j < j + 1 < l \leq n. \quad (4)$$

Below, we use another characterization of $\mathcal{D}$ and $\mathcal{K}$ which is formulated in the following proposition.

**Proposition 2.5:** A symmetric $n \times n$ matrix $C$ is a Demidenko matrix iff

$$\max_{1 \leq i \leq j - 1} \{c_{ij} - c_{i,j+1}\} \leq \min_{j+2 \leq i \leq n} \{c_{j,l} - c_{j+1,l}\} \quad \text{for all } 2 \leq j \leq n - 2. \quad (5)$$
A symmetric $n \times n$ matrix $C$ is a Kalmanson matrix iff

$$c_{i,j+1} + c_{i+1,j} \leq c_{ij} + c_{i+1,j+1} \quad \text{for all } 1 \leq i \leq n-2, \ i+2 \leq j \leq n-1$$

(6)

$$c_{i,j} + c_{i+1,n} \leq c_{n} + c_{i+1,1} \quad \text{for all } 2 \leq i \leq n-2.$$  \hspace{1cm} (7)

**Observation 2.6:** For a symmetric $n \times n$ matrix $C$, it can be decided in $O(n^2)$ time whether $C$ is a Kalmanson matrix (Demidenko matrix, respectively).

**Proof:** Characterization (5) for Demidenko matrices and characterization (6) and (7) for Kalmanson matrices both can be verified in $O(n^2)$ time. ■

3. GEOMETRIC PRELIMINAIRES AND DEFINITIONS

This section deals with planar Euclidean point sets whose distance matrices are permuted Demidenko, Kalmanson or Supnick matrices. Let $P = \{v_1, v_2, \ldots, v_n \subseteq \mathbb{R}^2$ be a sequence of points in the Euclidean plane and let $C$ denote its distance matrix defined by $c_{ij} = d(v_i, v_j)$ where $d(x, y)$ denotes the Euclidean distance between points $x$ and $y$. If the distance matrix $C$ fulfills the Demidenko (Kalmanson, Supnick) conditions, it is called a **Euclidean Demidenko** (Kalmanson, Supnick) matrix the sequence $P$ is called a **Demidenko** (Kalmanson, Supnick) point sequence, and the points in $P$ are said to form a **Demidenko** (Kalmanson, Supnick) point set. A permutation of $P$ that transforms the distance matrix into a Demidenko (Kalmanson, Supnick) matrix is called a **Demidenko** (Kalmanson, Supnick) permutation for $P$. For any rearranged subsequence $P'$ of the points in $P$, we denote by $\diamondsuit P'$ the sequence of indices in $P'$.

![Figure 1. A Kalmanson point set.](image)

Recherche opérationnelle/Operations Research
For \( x, y \in \mathbb{R}^2 \) and \( \Delta \in \mathbb{R} \), denote by \( h(x, y, \Delta) = \{ p \in \mathbb{R}^2 | d(x, p) - d(y, p) = \Delta \} \) the set of points \( p \in \mathbb{R} \) which lie on one (uniquely determined) branch of the hyperbola with focal points at \( x \) and \( y \) and by \( H(x, y, \Delta) = \{ p \in \mathbb{R}^2 | d(x, p) - d(y, p) \geq \Delta \} \) the set of points \( p \in \mathbb{R}^2 \) in the infinite region bounded by \( h(x, y, \Delta) \) that does not contain the focal point \( x \). Finally, define \( \Delta^k_i = d(v_{k-1}, v_i) - d(v_k, v_i) \) for \( 2 \leq k \leq n \).

**Theorem 3.1:** A point sequence \( P = (v_1, \ldots, v_n) \) is a Demidenko point sequence if and only if for each, \( 4 \leq p \leq n \), the point \( v_p \) lies within the region

\[
H_p = \bigcap_{k=3}^{p-1} H(v_{k-1}, v_k, \Delta^k_i),
\]

where \( \Delta^k = \max\{\Delta^k_i | i = 1, \ldots, k - 2\} \).

**Proof:** The proof is done by induction on \( p \geq 4 \). For \( p = 4 \), condition (1) must be satisfied, i.e. \( v_4 \) must be located such that the relation \( d(v_2, v_1) + d(v_3, v_4) \leq d(v_2, v_4) + d(v_1, v_3) \) holds. This inequality is equivalent to \( v_4 \in H(v_2, v_3, \Delta^3) \) with \( \Delta^3 = d(v_2, v_1) - d(v_2, v_1) \).

Next, assume that the statement is true up to \( p - 1 \) and that the point sequence \( (v_1, \ldots, v_{p-1}) \) is a Demidenko point sequence. Then we only have to deal with those inequalities where point \( v_p \) is involved. By Observation 2.4, it is sufficient to show that condition (4) is fulfilled, i.e. that \( d(v_i, v_i) - d(v_{i+1}, v_i) \leq d(v_i, v_p) - d(v_{i+1}, v_p) \) for all \( i \) and \( j \) with \( 1 \leq i < j + 1 \leq p - 1 \) is equivalent to \( v_p \in H_p \). Let \( k = j + 1 \). Then \( d(v_{k-1}, v_i) - d(v_k, v_i) \leq d(v_{k-1}, v_p) - d(v_k, v_p) \) is equivalent to \( v_p \in H(v_{k-1}, v_k, \Delta^k_i) \). Since \( H_p \) is the intersection of all \( H(v_{k-1}, v_k, \Delta^k_i) \) for \( k = 3, \ldots, p - 1 \) and \( i = 1, \ldots, k - 2 \), the theorem follows.

In the geometric interpretation, conditions (1) and (2) both correspond to hyperbolas. Taking into account the characterization of \( K \) in Proposition 2.5, Kalmanson point sequences may be characterized in analogy to the above theorem.

**Theorem 3.2:** A point sequence \( P = (v_1, \ldots, v_n) \) is a Kalmanson point sequence if and only if it is a Demidenko point sequence and if each point \( v_p \in P, p \geq 4 \), belong to the region

\[
H'_p = \bigcap_{k=3}^{p-1} \bigcap_{i=1}^{k} H(v_{i+1}, v_i, -\Delta^i_{k+1}).
\]
Figure 1 gives an illustration for Kalmanson point sequences. Note the depicted point set is "almost convex" and that the numbering follows the "almost convex hull". All Kalmanson point sets that we constructed in our computation experiments had a similar shape. Figure 2 depicts a Demidenko point sequence. The optimum TSP tour for this point set is \(1, 2, 3, 4, 5, 6, 7, 8, 15, 14, 13, 12, 11, 10, 9\).

A point set \(P\) is called degenerate if all points in \(P\) lie on a common line and non-degenerate otherwise. A point set \(P\) is called convex if each of its points lies on the boundary of the convex hull. A sequence of points is called cyclically ordered, if its points form a convex set and if the numbering corresponds to the clockwise or counterclockwise order along the convex hull. In the case of a degenerated set, a cyclic ordering is one of the two orderings along the line.

**Observation 3.3:** Assume that the points \(v_1, v_2, v_3\) and \(v_4\) (in this order) form a non-degenerate convex quadrangle. Then

(i) \(d(v_1, v_3) + d(v_2, v_4) \geq d(v_1, v_2) + d(v_3, v_4)\) and \(d(v_1, v_3) + d(v_2, v_4) \geq d(v_2, v_3) + d(v_1, v_4)\) (i.e. the total length of the diagonals is greater than the total length of two opposite sides).

(ii) Up to cyclic shifts, \((v_1, v_2, v_3, v_4)\) and \((v_4, v_3, v_2, v_1)\) are the only permutations that yield Kalmanson sequences.

The following proposition is an easy consequence of Observation 3.3(i) above.

**Proposition 3.4:** (Kalmanson [7], folklore). If \(P = (v_1, \ldots, v_n) \subseteq \mathbb{R}^2\) is a non-degenerate, convex, cyclically ordered sequence of points, then its distance matrix is a Kalmanson matrix. Moreover, up to cyclic shifts, this
and its reverse permutation are the only orderings for the points in $P$ that yield Kalmanson sequences.

In case the Euclidean coordinates of all $n$ points of a convex set are explicitly given, a cyclic ordering (and thus a numbering that makes the point set a Kalmanson sequence) can be found in $O(n \log n)$ time by applying a standard convex hull algorithm (see e.g. Preparata and Shamos [10]). In case the coordinates of the points are not given explicitly, but only implicitly via the distance matrix, numerical and computational difficulties arise: In order to compute the exact coordinates from the distances, computations with irrational numbers are to be performed. This will lead to rounding errors and to numerical instabilities. Moreover, the computational standard models (Turing machine, random access machine) cannot cope with irrational numbers. For these reasons, all algorithms in this paper will be designed in such a way that they work directly with the distance matrix and without intermediate computation of Euclidean coordinates.

**Lemma 3.5:** For the Euclidean distance matrix of a convex point set $P$, the index sequence of a cyclic ordering of the points in $P$ can be computed in $O(n \log n)$ time without intermediate computation of Euclidean coordinates.

**Proof:** The cyclic ordering is easy to find if one has two adjacent points $x$ and $y$ on the convex hull. One can check that $d(x, v) - d(y, v)$ must not decrease as we visit the points $v$ by walking on the hull from $x$ to $y$ (the difference may remain constant for some time, for points in $P$ on the line through $x$ and $y$, but else it increases). Therefore, the correct ordering can be found by sorting. In order to find $x$ and $y$, we start with two arbitrary points $x$ and $z$ and select $y \in P \setminus \{x\}$ so that $d(x, y) - d(z, y)$ becomes minimum.

**Lemma 3.6:** If all points of a Euclidean point set $P$ lie on a common line, then the distance matrix of $P$ is a permuted Demidenko, Kalmanson and Supnick matrix.

**Proof:** Verify that if the points are sorted along the line, then the resulting distance matrix fulfills all conditions (1), (2), and (3).

4. PERMUTED EUCLIDEAN KALMANSON MATRICES

This section deals with the problem of recognizing permuted Euclidean Kalmanson matrices. For our purposes, the most important case of Kalmanson point sequences consists of two points $v_1$ and $v_n$ and $n - 2$ points lying...
on some hyperbola branch \( h(v_1, v_n, \Delta) \). The following two lemmas deal with this case.

**Lemma 4.1:** Let \( P = \{v_1, v_2, \ldots, v_n\} \) be a Kalmanson sequence for which all points, \( v_i, 2 \leq i \leq n - 1 \), lie on \( h(v_1, v_n, \Delta) \). If \( \Delta \geq 0 \), then the points in \( P \setminus \{v_1\} \) form a convex set and if \( \Delta \leq 0 \), the points in \( P \setminus \{v_n\} \) form a convex set.

**Proof:** We only deal with \( \Delta \geq 0 \), since the other case is symmetric. Hence, let \( \Delta \geq 0 \) and suppose that \( P \setminus \{v_1\} \) is not a convex set. Let \( V_1 \) contain those points of \( P \) which lie above or on the line \( L \) through \( v_1 \) and \( v_n \), and let \( V_2 \) contain those points below \( L \). Since \( P \setminus \{v_1\} \) is not convex, \( V_1 \neq \emptyset \). Let \( v_a \in V_1 \) and \( v_b \in V_2 \) be the points at minimum distance to \( L \). The line through \( v_a \) and \( v_b \) crosses the line segment connecting \( v_1 \) to \( v_n \) (otherwise \( P \setminus \{v_1\} \) would be convex). This yields that \( v_1, v_a, v_n \) and \( v_b \) (in this order) form a convex quadrangle and contradict Observation 3.3 (ii).

**Lemma 4.2:** Let \( P = \{v_1, v_2, \ldots, v_n\} \) be a point set for which all points \( v_i, 2 \leq i \leq n - 1 \), lie on \( h(v_1, v_n, \Delta) \). Then there exist at most two Kalmanson permutations for \( P \) that have \( v_1 \) as first point and \( v_n \) as last point. These two permutations can be computed in \( O(n \log n) \) time.

**Proof:** Lemma 4.1 yields that in case a Kalmanson permutation with the stated properties exists, then \( \{v_2, \ldots, v_{n-1}\} \) forms a convex set together with, say, point \( v_1 \). By Proposition 3.4, the only orderings that turn a convex set into a Kalmanson sequence, are the clockwise and counterclockwise orderings along the convex hull and cyclic shifts of these permutations. Since \( v_1 \) is the first point in the sequence, the cyclical ordering is anchored at \( v_1 \) and thus fixed up to orientation. Lemma 3.5 yields the time bound.

Next, a polynomial time recognition algorithm for permuted Euclidean Kalmanson matrices will be designed in two phases. In the first phase, we investigate the special case where the index \( p \) of the first point and the index \( q \) of the last point in the Kalmanson permutation are \textit{a priori} known. The second phase treats the general problem without any restrictions.

**Lemma 4.3:** Let \( C \) be the Euclidean distance matrix of some planar point set \( P = \{v_1, \ldots, v_n\} \) and let \( v_p \) and \( v_q \) be two points in \( P \). Then it can be decided in \( O(n^2) \) time whether there is a Kalmanson permutation that has \( v_p \) as first point and \( v_q \) as last point.

Recherche opérationnelle/Operations Research
Proof: The algorithm is mainly based on the above Lemmata 4.1 and 4.2, and it uses the fact that every cyclic shift of a Kalmanson sequence again is a Kalmanson sequence (cf. Observation 2.2). The algorithm consists of the following five Steps (A1)-(A5).

(A1) Compute \( \Delta(v) = d(v_p, v) - d(v_q, v) \) for all points \( v \) in \( P \setminus \{v_p, v_q\} \), and sort them by increasing \( \Delta(v) \) values. By grouping points with identical \( \Delta \)-values together, a partition of the set into \( m \leq n - 2 \) subsets \( P_i \) obtained, \( 1 \leq i \leq m \), such that all points in \( P_i \) have the same \( \Delta(v) \) value \( \Delta_i \) and \( \Delta_i < \Delta_{i+1} \) for \( 1 \leq i \leq m - 1 \).

Since \( v_p \) and \( v_q \) are the first and the last point in the Kalmanson point sequence, \( d(v_p, v) - d(v_q, v) \leq d(v_p, w) - d(v_q, w) \) must hold for all points \( v \) preceding point \( w \) just to fulfill condition (1). Hence, each set \( P_i \) must precede set \( P_{i+1} \) in a Kalmanson sequence, and the set of potentially feasible permutations is described by \( S_{TR}(\{p\}, \diamond P_1, \ldots, \diamond P_m, \{q\}) \)

(A2) For every set \( P_i \) with \( s = |P_i| > 1, 1 \leq i \leq m \) do: if \( \Delta_i \leq 0 \), construct a cyclic ordering \( \sigma_i' \) of the points \( P_i \cup \{v_p\} \), otherwise construct a cyclic ordering \( \sigma_i' \) of the points \( P_i \cup \{v_q\} \). This yields a permutation \( \sigma_i' = \langle p, x_1, \ldots, x_s \rangle \) or \( \sigma_i' = \langle x_1, \ldots, x_s, q \rangle \) of the indices of the points in \( P_i \). Set \( \sigma_i'' = \langle x_1, \ldots, x_s \rangle \).

If \( m = 1 \), compute two permutations according to Lemma 4.2. Check whether one of them indeed yields a Kalmanson sequence. Stop.

Note that every set \( P_i \) is located on the branch of a hyperbola. Lemma 4.1 yields that for every \( i, P_i \cup \{v_p\} \) or \( P_i \cup \{v_q\} \) is a convex set (depending on the sign of \( \Delta_i \)). Similarly as in Lemma 4.2 this implies that for every such convex set the only orderings that turn the set into a Kalmanson sequence, are the clockwise and counterclockwise orderings along the convex hull. These orderings are computed (up to orientation) in Step (A.2), and it remains to determine the right orientation for every ordering.

(A3) For every permutation \( \sigma_i'' = \langle x_1, \ldots, x_s \rangle \) of a set \( P_i \) with \( |P_i| = s > 1 \) do: compute the value \( \Psi_i = d(v_p, v_{x_1}) - d(v_p, v_{x_s}) \).

If \( d(v, v_{x_1}) - d(v, v_{x_s}) = \Psi_i \) for all \( v \in P \setminus P_i \), then find two permutations for \( P \setminus P_i \cup \{v_{x_1}, v_{x_s}\} \) as in Lemma 4.2. In both permutations, replace the sequence \( x_1, x_s \) by \( \sigma_i'' \) (respectively, \( x_s, x_1 \) by \( (\sigma_i'')^{-} \)). Check whether one of them indeed yields a Kalmanson sequence and whether it (or one of its cyclic shifts) has \( v_p \) and \( v_q \) as first and last point. Stop.

vol. 31, n° 4, 1997
Consider the branch \( h(v_{x_1}, v_{x_s}, \Psi_i) \). It contains \( v_p \) by definition and it is not hard to see that it also contains \( v_q \). In case this branch also covers all other points in \( P \setminus P_i \), Lemma 4.1 applies to the set \( P \setminus P_i \cup \{v_{x_1}, v_{x_s}\} \). We know that in any feasible Kalmanson sequence, \( P_i \) is a contiguous subsequence and hence we may replace the two indices \( x_1 \) and \( x_s \) by an appropriate cyclic ordering of \( P_i \).

(A4) Otherwise, there exists some point \( v \) with \( d(v, v_{x_1}) - d(v, v_{x_s}) \neq \Psi_i \).

If \( v \in P_1 \cup \ldots \cup P_{i-1} \) and \( d(v, v_{x_1}) - d(v, v_{x_s}) < \Psi_i \) or if \( v \in P_{i+1} \cup \ldots \cup P_m \) and \( d(v, v_{x_1}) - d(v, v_{x_s}) > \Psi_i \), then \( x_1 < x_s \) in \( \sigma_i \) and otherwise \( x_s < x_1 \) in \( \sigma_i \). Set \( \sigma_i = \sigma_i'' \) or \( \sigma_i = (\sigma_i'')^{-} \), depending on the relative placement of \( x_1 \) and \( x_s \).

(A5) For every \( P_i \) of cardinality one, \( P_i = \{v_z\} \), define \( \sigma_i = (z) \). Compute \( \sigma \) by gluing together \( p, \sigma_1, \ldots, \sigma_m, q \). Test if \( C_\sigma \in K \). Stop.

If the algorithm branches into (A4), then there exists some points \( v \notin h(v_{x_1}, v_{x_s}, \Psi_i) \). Assume without loss of generality that \( v \in P_1 \cup \ldots \cup P_{i-1} \) and that \( d(v, v_{x_1}) - d(v, v_{x_s}) < \Psi_i \) (all other cases are symmetric). The problem boils down to deciding whether the ordering \( \langle v_p, v, v_{x_s}, v_{x_1}, v_q \rangle \) or whether \( \langle v_p, v, v_{x_1}, v_{x_s}, v_q \rangle \) is the correct ordering. Since \( d(v, v_{x_1}) - d(v, v_{x_s}) < \Psi_i = d(v_p, v_{x_1}) - d(v_p, v_{x_s}) \), the second ordering contradicts condition (2). Thus, it is infeasible and \( v_{x_1} \) must precede \( v_{x_s} \). Exactly this check is performed in Step (A4).

Finally, in Step (A5) the orderings for the sets \( P_i \) are composed to a potential solution permutation \( \sigma \). Since \( \sigma \) was computed just by investigating necessary conditions, we must verify in the end whether it indeed yields a Kalmanson sequence.

The correctness of the algorithm is clear by the above arguments, and it remains to prove the claimed time complexity. The sorting and grouping in Step (A1) is done in \( O(n \log n) \) time. Computing the orderings along the convex hulls of all \( m \) sets \( P_i \) in Step (A2) is done in overall time \( O(n \log n) \) by applying the algorithm described in Lemma 3.5. The case \( m = 1 \) is handled according to Lemma 4.2 in \( O(n \log n) \) time. Steps (A3) and (A4) together can at most \( O(n) \) time per set \( P_i \) and thus are performed in \( O(n^2) \) time. By Observation 2.6, testing permutation \( \sigma \) in Step (A5) takes \( O(n^2) \) time. Summarizing, this yields an overall time complexity of \( O(n^2) \) and the proof of Lemma 4.3 is complete.

\textbf{Theorem 4.4:} For the \( n \times n \) distance matrix \( C \) of a Euclidean point set \( P \), it can be decided in \( O(n^2) \) time whether \( C \) is a permuted Kalmanson matrix.
Proof: Applying the algorithm in Lemma 4.3 to each of the $O(n^2)$ pairs of indices $(p, q)$ yields a naive $O(n^4)$ time algorithm for recognizing permuted Euclidean Kalmanson matrices. To improve on this, we generate a small (constant size) set $S$ of candidate pairs with the following property: in case $C$ is a permuted Euclidean Kalmanson matrix, then there exists a Kalmanson sequence of $P$ in which at least one of the pairs in $S$ is adjacent. Then we call the algorithm designed in Lemma 4.3 for every index pair $(p, q) \in S$.

By the definition of $S$, the procedure will succeed for at least one pair and yield a permutation that transforms $C$ into a Kalmanson matrix.

Hence, it remains to explain how to generate the constant size set $S$ in at most $O(n^2)$ time: choose two arbitrary indices $i$ and $j$ and compute the set $M(i, j)$. If $|M(i, j)| \leq 2$, then let $S$ contain all pairs over $M(i, j) \cup \{i\}$. If $|M(i, j)| \geq 3$, observe that all points with index in $M(i, j)$ lie on $h(v_i, v_j, \Delta)$ for some appropriate $\Delta$ and thus form a convex set. Compute the indices $k, l$ and $m$ of three consecutive points on the hull and let $S$ contain all pairs over $\{k, l, m\}$.

By Lemma 2.3, there exists a cyclic shift that makes the points corresponding to $M(i, j) \cup \{i\}$ a prefix of some Kalmanson sequence. This justifies the definition of $S$ in case $|M(i, j)| \leq 2$ holds. If $|M(i, j)| \geq 3$ holds, then the ordering of this convex set within a Kalmanson sequence must follow the convex hull (cf. Observation 3.4) and thus is fixed up to orientation and up to cyclic shifts. By Lemma 2.3, there exists a Kalmanson permutation that has a prefix this convex set with the point $v_i$ somewhere inbetween. Out of three consecutive points on the hull, at most one pair can be separated by the point $v_i$ in the Kalmanson sequence.

5. PERMUTED EUCLIDEAN DEMIDENKO MATRICES

This section deals with the recognition of permuted Euclidean Demidenko matrices. Our approach is conceptually similar to the approach for Euclidean Kalmanson matrices described in the preceding section. The main difference (and main difficulty) arises from the fact that condition (2) need not be fulfilled by Demidenko matrices. Hence, less combinatorial structure is imposed. e.g. Lemmata 4.1 and 4.2 are not necessarily true for Demidenko point sequences and (worst of all!) cyclic shifts of Demidenko permutations do not necessarily yield Demidenko permutations.

For an Euclidean point set $P$ two points $f_1, f_2 \in P$ are called a pair of focal points for $P$, if there exists a real $\Delta$ such that all other points in $P \setminus \{f_1, f_2\}$ lie on $h(f_1, f_2, \Delta)$. We will make use of the following
two observations (where the first observation is elementary and the second observation is an easy consequence of the first one).

**Observation 5.1:** Two branches of two not-identical hyperbolas intersect in at most four points. ■

**Observation 5.2:** A set \( P \) of \( n \geq 9 \) points in the Euclidean plane possesses at most one pair \( \{ f_1, f_2 \} \) of focal points for \( P \). ■

The polynomial time recognition algorithm is designed in three phases. In the first phase, we deal with the special case where (i) the index \( p \) of the first point and the index \( q \) of the last point in the Demidenko permutation are a priori known and where (ii) \( v_p \) and \( v_q \) are not a pair of focal points for the underlying point set \( P \). This forms the main part of this section. The second phase treats the complementary case where \( v_p \) and \( v_q \) are a pair of focal points for \( P \). Finally, in the third phase the general problem without any restrictions is solved.

**Lemma 5.3:** Let \( C \) be the Euclidean distance matrix of some planar point set \( P = \{ v_1, \ldots, v_n \} \), let \( v_p \) and \( v_q \) be two points in \( P \) that are not a pair of focal points for \( P \). Then it can be decided in \( O(n^2) \) time whether there is a Demidenko permutation that has \( v_p \) as first point and \( v_q \) as last point.

**Proof:** We will call a Demidenko permutation that has \( v_p \) as first point and \( v_q \) as last point an appropriate Demidenko permutation. The algorithm consists of three STEPS (B1), (B2), and (B3). Recall that Step (A1) in the preceding section only exploited the Demidenko condition (1). Hence, we may start the same way and have (B1) identical to (A1).

(B1) Compute \( \Delta(v) = d(v_p, v) - d(v_q, v) \) for all points \( v \) in \( P \setminus \{ v_p, v_q \} \), and sort them by increasing \( \Delta(v) \) values. By grouping points with identical \( \Delta \)-values together, a partition of the set into \( m \leq n - 2 \) subsets \( P_i \) is obtained, \( 1 \leq i \leq m \), such that all points in \( P_i \) have the same \( \Delta(v) \) value \( \Delta_i \) and \( \Delta_i < \Delta_{i+1} \) for \( 1 \leq i \leq m - 1 \).

Again in any appropriate Demidenko permutation, set \( P_i \) must precede set \( P_{i+1} \). Hence all appropriate Demidenko permutations are contained in \( S_{TR_1} = S_{TR}(\{p\}, \Diamond P_1, \ldots, \Diamond P_m, \{q\}) \). Moreover, subset \( P_i \) is situated on the hyperbola branch \( h(v_p, v_q, \Delta_i) \) with \( \Delta_i = \Delta(v) \) for \( v \in P_i \). Note that \( m \geq 2 \), since \( v_p \) and \( v_q \) are not focal points for \( P \). Consequently, \( P_1 \neq P_m \).

Next, select two arbitrary points \( v_r \in P_1 \) and \( v_s \in P_m \). In any appropriate Demidenko permutation, point \( v_r \) precedes all points in \( P_2 \cup \ldots \cup P_{M-1} \cup \{ v_q \} \).
Analogously to Step (B1), sort the differences \( d(v, v_r) - d(v, v_q) \) for \( v \in \bigcup_{i=2}^{m} P_i \) increasingly and obtain another partition \( R_1, \ldots, R_{\mu} \) by grouping points with identical values together. Symmetrically, point \( v_s \) comes after all points in \( \{v_p\} \cup P_1 \cup \ldots \cup P_{m-1} \). Sorting the differences \( d(v, v_p) - d(v, v_s) \) increasingly for all \( v \in \bigcup_{i=1}^{m-1} P_i \) results in a third partition \( S_1, \ldots, S_{\nu} \) of \( \bigcup_{i=1}^{m-1} P_i \). The way we derived the three partitions implies that any appropriate Demidenko is contained in

\[
S_{TR_1} \cap S_{TR}(\{p\}, \diamond P_1, \diamond R_1, \ldots, \diamond R_{\mu}, \{q\}) \\
\cap S_{TR}(\{p\}, \diamond S_1, \ldots, S_{\nu}, \diamond P_m, \{q\}).
\]

These explanations clarify and justify the next Step (B2).

(B2) Select \( v_r \in P_1 \) and \( v_s \in P_m \). Compute \( \Delta^1(v) = d(v, v_r) - d(v, v_q) \) for all points \( v \) in \( \bigcup_{i=2}^{m} P_i \) and construct the partition \( R_1, \ldots, R_{\mu} \) by sorting and grouping the points according to their \( \Delta^1\)-values. Set

\[
S_{TR_2} = S_{TR}(\{p\}, \diamond P_1, \diamond R_1, \ldots, \diamond R_{\mu}, \{q\}).
\]

Compute \( \Delta^1(v) = d(v, v_p) - d(v, v_s) \) for all points \( v \) in \( \bigcup_{i=1}^{m-1} P_i \) and construct the partition \( S_1, \ldots, S_{\nu} \) by sorting and grouping the points according to their \( \Delta^2\)-values. Set

\[
S_{TR_3} = S_{TR}(\{p\}, \diamond P_1, \diamond S_1, \ldots, \diamond S_{\nu}, \{q\}).
\]

Compute \( S_{TR} = S_{TR_1} \cap S_{TR_2} \cap S_{TR_3} \). In case \( S_{TR} \) is empty, stop with the answer "NO APPROPRIATE PERMUTATION EXISTS". Otherwise, there is a partition \( T_1, \ldots, T_\kappa \), of \( P \) with \( T_1 = \{v_p\}, T_\kappa = \{v_q\} \), and \( |T_i| \leq 4 \) for all \( 2 \leq i \leq \kappa - 1 \) such that \( S_{TR} = S_{TR}(\diamond T_1, \ldots, \diamond T_\kappa) \).

Intersecting \( S_{TR_1}, S_{TR_2}, \) and \( S_{TR_3} \) is done according to Proposition 2.1 (this proposition also guarantees the existence of stripes \( T_i \)). Since the points of every \( T_i \) are intersection points of at least two non-identical hyperbola branches (the hyperbolas have distinct focal points), Observation 5.1 yields \( |T_j| \leq 4 \) for all \( 1 \leq j \leq \kappa \). Hence, all that remains to do is to determine the internal orderings in every set \( T_i \). Recall that by condition (5) for two neighboring points \( p_j \) and \( p_{j+1} \) in a Demidenko sequence

\[
\max_{1 \leq i \leq j-1} \{c_{ij} - c_{i,j+1}\} \leq \min_{j+2 \leq l \leq n} \{c_{j,l} - c_{j+1,l}\} \quad (8)
\]

must holds. Conversely, if (8) for all neighboring points \( p_j \) and \( p_{j+1} \) with \( 2 \leq j \leq n - 2 \) than this ordering indeed is a Demidenko ordering.

Now consider some fixed permutation \( \pi \) of the elements of some set \( T_i \) with \( |T_i| \geq 2 \). We test for every pair of neighboring indices \( x \) and \( y \) in this permutation (where \( x \) comes before \( y \)) whether they fulfill the inequality corresponding to (8) as follows: let \( Q_1 \) contain all numbers in \( T_1 \cup \ldots \cup T_{i-1} \)

\[
\max_{1 \leq i \leq j-1} \{c_{ij} - c_{i,j+1}\} \leq \min_{j+2 \leq l \leq n} \{c_{j,l} - c_{j+1,l}\} \quad (8)
\]

must holds. Conversely, if (8) for all neighboring points \( p_j \) and \( p_{j+1} \) with \( 2 \leq j \leq n - 2 \) than this ordering indeed is a Demidenko ordering.
together with all numbers in $T_i$ that precede $x$ and $y$ in $\pi$ (if any exist). Let $Q_2$ contain all numbers in $T_{i+1} \cup \ldots \cup T_K$ together with all numbers in $T_i$ that succeed $x$ and $y$ in $\pi$. The necessary condition to hold is

$$\max_{k \in Q_1} \{c_{kx} - c_{ky}\} \leq \min_{l \in Q_2} \{c_{xl} - c_{yl}\}.$$ 

The permutation $\pi$ is called a nice permutation for $T_i$ if all of its pairs of neighboring indices pass this test. For sets $T_i$ with $|T_i| = 1$, the unique possible (trivial) permutation is nice by definition.

A similar test is performed for every index $x$ in $T_{i-1}$ and every index $y$ in $T_i$ (with $Q_1 = (T_1 \cup \ldots \cup T_{i-1}) \setminus \{x\}$) and $Q_2 = (T_1 \cup \ldots \cup T_K) \setminus \{y\}$). In cases the indices $x$ and $y$ pass this test, they are called nicely adjacent.

(B3) Construct a directed auxiliary graph $G = (V, E)$: for any nice permutation for any set $T_i$ with $|T_i| \geq 2$, there is a corresponding vertex in $V$. If $\pi_1$ is nice for $T_{i-1}$, $\pi_2$ is nice for $T_i$ and if the last element in $\pi_1$ is nicely adjacent to the first element in $\pi_2$, then there is an edge in $E$ going from the vertex corresponding to $\pi_1$ to the vertex corresponding to $\pi_2$.

Test whether in $G$ there is a directed path going from the (unique) vertex corresponding to $T_1$ to the (unique) vertex corresponding to $T_K$. $G$ is a permuted Demidenko matrix if an only if such a path exists. In case the path exists, a solution permutation can be computed by concatenating all nice permutations along this path.

It is easy to see that the existence of appropriate Demidenko is equivalent to the existence of a connecting path in the auxiliary graph $G$: in $G$ there are only edges going from permutations corresponding to $T_{i-1}$ to permutations corresponding to $T_i$. Because of this leveled structure, any path connecting $T_1$ to $T_K$ in $G$ must visit exactly one nice permutation for every $T_i$. Hence, it spans the whole set $P$. By the definition of "nice" and "nicely adjacent", every pair of adjacent indices along this path fulfills condition (8) and hence, by Proposition 2.5, the corresponding permutation is a Demidenko permutation. On the other hand any Demidenko permutation trivially gives rise to a path connecting $T_1$ to $T_K$.

It remains to analyze the time complexity of the above algorithm. The sorting in Step (B1) costs $O(n \log n)$ time, the grouping operations are done in $O(n)$ time. Analogously, computing $S_{TR_2}$ and $S_{TR_3}$ in (B2) costs $O(n \log n)$ time. According to Proposition 2.1, intersecting the three sets of permutations is performed in linear time. The auxiliary graph in Step (B3)
has at most $24\kappa$ vertices and it is easy to verify that the number of edges is also $O(\kappa)$. Testing whether a permutation is nice for some $T_i$ and whether two indices are nicely adjacent amounts to computing the minimum and maximum of two sets with $O(n)$ elements according to (8). Hence, $G$ can be constructed in $O(n\kappa)$ time. Testing for the existence of a connecting path is solved e.g. by Depth-First-Search in time linear in the number of edges and vertices in a graph. Hence, $O(n)$ time is sufficient for this. Since $\kappa \leq n$, the overall time complexity is $O(n^2)$. This completes the proof of Lemma 5.3.

**Lemma 5.4:** Let $C$ be the Euclidean distance matrix of some planar point set $P = \{v_1, \ldots, v_n\}$, let $v_p$ and $v_q$ be a pair of focal points for $P$. Then it can be decided in $O(n^3)$ time whether there is a Demidenko permutation that has $v_p$ as first point and $v_q$ as last point.

**Proof:** There are $n - 2$ candidates for the second point $p_x$ in a Demidenko permutation. For a fixed candidate point $p_x$, all appropriate Demidenko permutations are in $S_{TR}(\{p\}, \{x\}, I\{p, x, q\}, \{q\})$. Hence, we are in a situation analogous to that one after Step (B1) in the algorithm in the preceding Lemma 5.3 (i.e. $m \geq 2$ holds and $P_1 \neq P_m$). Performing Steps (B2) and (B3) in $O(n^2)$ time per candidate results in $O(n^3)$ overall time as claimed above.

**Theorem 5.5:** For the distance matrix $C$ of some Euclidean point set, it can be decided in $O(n^4)$ time whether $C$ is a permuted Demidenko matrix.

**Proof:** For $n \leq 8$ check all possible permutations of $C$ in constant time. For $n \geq 9$, test for every pair $v_p, v_q \in P$ whether there is an appropriate Demidenko permutation with $v_p$ as first point and $v_q$ as last point. By Lemmata 5.3 and 5.4, this takes $O(n^2)$ time for every non-focal pair of points and $O(n^3)$ for focal pairs of points. Since by Observation 5.2, there is at most one pair of focal points for $P$ the claimed overall time complexity $O(n^4)$ follows.

6. **Permutated Euclidean Supnik Matrices**

In this section, it will be shown that the combinatorial structure of Supnik point sets is rather primitive: in case a Supnik set contains $n \geq 9$ points, all these points must lie on a common straight line. Hence, Supnik point sets are trivial to recognize. This result was also mentioned without proof.

**Proposition 6.1:** Any non-degenerate point set $P$ in the Euclidean plane with $|P| \geq 9$ contains a non-degenerate subset $P^*$ such that (i) $|P^*| = 5$ and (ii) $P^*$ is a convex set.  

**Proposition 6.2:** Let $C$ be a $5 \times 5$ Supnick matrix. Then the our $(1, 3, 5, 4, 2)$ yields a shortest travelling salesman tour.

A proof for Proposition 6.1 can be found in the book by Lovász [9] (solution to problem 15.31). Proposition 6.2(b) follows from Supnick’s result [12] as described in the introduction section.

**Lemma 6.3:** Let $P$ be a Supnick point set with $|P| \geq 9$. Then all points in $P$ lie on a common line.

**Proof:** Suppose the contrary and let $(v_1, \ldots, v_n)$ be a numbering of $P$ such that the corresponding distance matrix is a Supnick matrix. $P$ fulfills the conditions of Proposition 6.1 and hence contains a convex non-degenerate subsets $P^*$ on five points, without loss of generality $P^* = \langle v_1, v_2, v_3, v_4, v_5 \rangle$. By Proposition 6.2, the induced ordering $(v_1, v_3, v_5, v_4, v_2)$ of the points in $P^*$ yields a shortest tour and obviously, this tour must follow the convex hull. This in turn implies that the points $v_5, v_4, v_2, v_1$ (in this order) form a convex quadrangle and also fulfill the Supnick condition $d(v_1, v_4) + d(v_2, v_5) \leq d(v_1, v_5) + d(v_2, v_4)$. This is a contradiction to Observation 3.3(i).

**Theorem 6.4:** For the distance matrix $C$ of some Euclidean point set, it can be decided in $O(n^2)$ time whether $C$ is a permuted Supnick matrix.

**Proof:** For $n \leq 8$, check all possible permutations whether they yield a Supnick matrix. For $n \geq 9$, check whether $C$ is the distance matrix of a point set on a line and apply Lemma 3.6.

**7. CONCLUSION AND OPEN PROBLEMS**

In this paper we have shown how to recognize in polynomial time Euclidean point sets whose distance matrices fulfill the Demidenko, Kalmanson, or Supnick condition for an appropriate numbering of the points. The applied methods heavily relied on geometric features of the problems and strongly exploited geometric properties like convexity.
Several related questions remain open: for which other "nice" classes of matrices is it polynomial time decidable whether the distance matrix of some given Euclidean point set belongs to this class? One potential candidate for such a nice class are the symmetric Van der Veen matrices [13] defined by

\[ c_{ij} + c_{kl} \leq c_{il} + c_{jk} \quad \text{for} \ 1 \leq i < j < k < l \leq n. \]

There is a geometric characterization of Euclidean Van der Veen point sequences via hyperbolas analogous to the characterization in Theorems 3.1 and 3.2 for Demidenko and Kalmanson point sequences. However, we did not succeed in finding a polynomial time recognition algorithm for Van der Veen point sets.

Another problem consists in deriving polynomial time algorithms for recognizing arbitrary permuted Demidenko, Kalmanson, and Supnick matrices (that do not necessarily result from Euclidean point sets). Without the geometric structures, such recognition problems clearly become much harder. A first step towards a solution was taken by Deineko, Rudolf and Woeginger [2] who showed how to recognize permuted \( n \times n \) Supnick matrices in \( O(n^2 \log n) \) time. Note that compared to the geometric case, this running time is a \( O(n \log n) \) factor slower.

ACKNOWLEDGEMENT

We would like to thank Bettina Klinz for carefully reading the paper and for many helpful comments.

REFERENCES

5. P. C. Gilmore, E. L. Lawler and D. B. Shmoys, Well-solved special cases, Chapter 4 in [8], pp. 87-143.

vol. 31, n° 4, 1997