



Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Travelling salesman paths on Demidenko matrices

Eranda Çela^{a,*}, Vladimir G. Deineko^b, Gerhard J. Woeginger^c

^a Department of Discrete Mathematics, TU Graz, Austria

^b Warwick Business School, University of Warwick, United Kingdom

^c Department of Computer Science, RWTH Aachen, Germany

ARTICLE INFO

Article history:

Received 28 September 2020

Received in revised form 15 September 2021

Accepted 21 November 2021

Available online xxx

Keywords:

Combinatorial optimization

Computational complexity

Travelling salesman problem

Tractable special case

Demidenko matrix

ABSTRACT

In the path version of the Travelling Salesman Problem (Path-TSP), a salesman is looking for the shortest Hamiltonian path through a set of n cities. The salesman has to start his journey at a given city s , visit every city exactly once, and finally end his trip at another given city t .

In this paper we show that a special case of the Path-TSP with a Demidenko distance matrix is solvable in polynomial time. Demidenko distance matrices fulfill a particular condition abstracted from the convex Euclidian special case by Demidenko (1979) as an extension of an earlier work of Kalmanson (1975). We identify a number of crucial combinatorial properties of the optimal solution and design a dynamic programming approach with time complexity $O(n^6)$.

© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The travelling salesman problem (TSP) is one of the best studied problems in operational research. This is not only due to the numerous appearances of the TSP in various practical applications, but also to its pivotal role in developing and testing new research methods. We refer the reader to the books by Applegate, Bixby, Chvátal & Cook [2], Gutin & Punnen [14], and Lawler, Lenstra, Rinnooy Kan & Shmoys [19] for a wealth of information on these issues. We also emphasize the role of the TSP in education and in the popularization of science. There is hardly any text book on operational research where the TSP would not be mentioned. The book [5] of Cook is an excellent example of how the TSP is used in the popularization of science.

An instance of the TSP consists of n cities together with an $n \times n$ symmetric distance matrix $C = (c_{ij})$ that specifies the distance c_{ij} between any pair i and j of cities. The objective in the TSP is to find a shortest closed route which visits each city exactly once; such a closed route is called a TSP tour. In the *Path-TSP* the instance also specifies two cities s and t (with $s \neq t$), and the goal is to find a shortest route starting at city s , ending at city t , and visiting all the other cities exactly once. Both the TSP and the Path-TSP are NP-hard to solve exactly (see for instance Garey & Johnson [12]), and both problems are APX-hard to approximate (Papadimitriou & Yannakakis [22]; Zenklusen [25]). These intractability results hold even in the metric case, where the distances between the cities are non-negative and satisfy the triangle inequality. In the non-metric case, where the distances between the cities do not fulfill the triangle inequality, the existence of a polynomial-time constant-factor approximation algorithm for the TSP would imply P=NP, see e.g. [24]. An analogous statement holds for the non-metric Path-TSP, e.g. by applying the result of Traub et al. [23]. Given the intractability of TSP and Path-TSP, the characterization of tractable special cases is of obvious interest and forms a well-established and vivid branch of research resulting in a number of publications [3,4,6,7,9–11,13,16–18].

* Corresponding author.

E-mail addresses: cela@opt.math.tu-graz.ac.at (E. Çela), Vladimir.Deineko@wbs.ac.uk (V.G. Deineko), woeginger@algo.rwth-aachen.de (G.J. Woeginger).

<https://doi.org/10.1016/j.dam.2021.11.019>

0166-218X/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1.1. Polynomially solvable cases of the TSP

The literature contains an impressive number of polynomially solvable cases for the classical TSP, as certified by the surveys of Burkard & al [3], Deineko, Klinz, Tiskin & Woeginger [6], Gilmore, Lawler & Shmoys [13], and Kabadi [16], and the references therein. We now briefly discuss three tractable cases of the TSP that are relevant for the current paper.

In the so-called *Convex-Euclidean TSP*, the cities are points in the Euclidean plane which are all located on the boundary of their convex hull. A folklore result says that the optimal tour in the Convex-Euclidean TSP is the cyclic walk along the convex hull, taken either in clockwise or in counter-clockwise direction. If we number the cities as $1, 2, \dots, n$ in clockwise order along the convex hull, then the cities satisfy the so-called *quadrangle inequalities*:

$$c_{ij} + c_{k\ell} \leq c_{j\ell} + c_{ik} \quad \text{for all } 1 \leq i < j < k < \ell \leq n \quad (1)$$

$$c_{i\ell} + c_{jk} \leq c_{j\ell} + c_{ik} \quad \text{for all } 1 \leq i < j < k < \ell \leq n \quad (2)$$

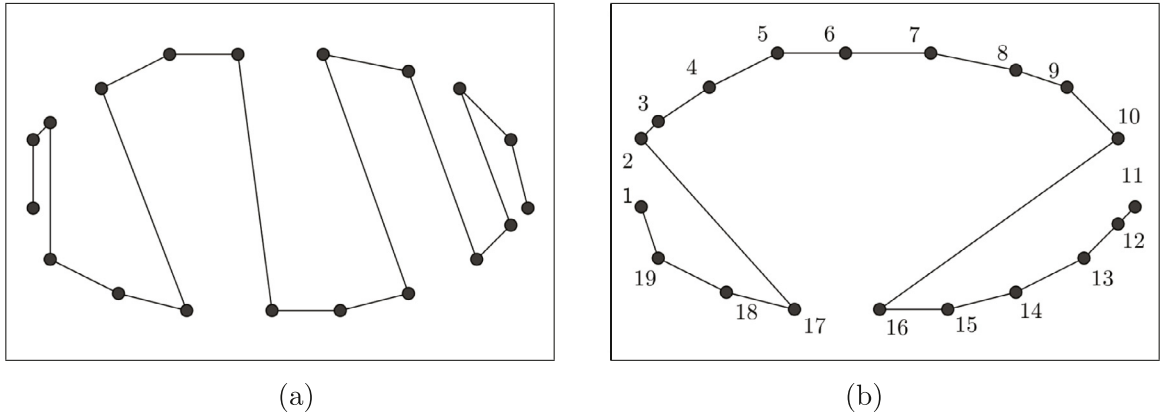
These quadrangle inequalities state the fact that in a convex quadrangle, the total length of two opposing sides is less than or equal to the total length of the two diagonals. Kalmanson [17] observed that whenever a TSP instance satisfies the quadrangle inequalities (1)–(2), the tour $1, 2, \dots, n$ is a shortest TSP tour. This extends the tractability of the Convex-Euclidean TSP to the tractability of the TSP on so-called *Kalmanson matrices*, where the distances satisfy (1)–(2). We stress that the class of Kalmanson distance matrices is large and goes far beyond the Convex-Euclidean case: Consider for instance a rooted ordered tree with non-negative edge lengths, place a city in each of the leaves, and number the cities from left to right. Then the shortest path distances c_{ij} between cities i and j determine a Kalmanson matrix, as the inequalities (1) and (2) can easily be verified for any quadruple of leaves. Finally, we mention the TSP on *Demidenko matrices*, where the distances just satisfy (1). Hence Demidenko matrices form an obvious generalization of Kalmanson matrices, and it is known that this generalization is proper. A famous result of Demidenko [9] shows that the TSP on Demidenko matrices is solvable in polynomial time. The gist of this paragraph is that the Convex-Euclidean TSP, the TSP on Kalmanson matrices, and the TSP on Demidenko matrices form three tractable TSP cases of strictly increasing generality.

The conditions (1) and (2) belong to the class of the so-called four-points conditions. [6] provides a detailed analysis of the computational complexity of 21 special cases of the TSP, where the distance matrix fulfills different types of four-points conditions (including Kalmanson and Demidenko matrices). A natural question in the context of investigations of four-points conditions is the corresponding recognition problem, described as follows. For a given distance matrix, find a permutation of the cities (i.e. a common permutation of the rows and the columns of the distance matrix) such that the resulting permuted matrix satisfies the considered four-points condition, or state that such a permutation does not exist. In the affirmative case we say that the distance matrix is a permuted matrix fulfilling the corresponding four-points condition (e.g. a permuted Kalmanson matrix or a permuted Demidenko matrix). For certain types of four-points conditions there are polynomial-time algorithms to solve the recognition problem, see [6]. In particular the recognition of permuted Kalmanson matrices is solvable in polynomial-time. Clearly, if the TSP (Path-TSP) is polynomially solvable for a certain class of distance matrices (e.g. Kalmanson matrices) and the recognition problem for permuted matrices of this particular class is solvable in polynomial time (as in the Kalmanson case), then the special case of the TSP (Path-TSP) with a permuted distance matrix from this class (e.g. permuted Kalmanson matrix) is also solvable in polynomial time. The optimal solution is obtained by composing the permutation output by the recognition algorithm with the optimal solution of the polynomially solvable case with the distance matrix fulfilling the corresponding four-points condition.

1.2. Polynomially solvable cases of the Path-TSP

Whereas the literature on polynomially solvable cases of the classical TSP forms a rich and comprehensive body, we are only aware of a single result on polynomially solvable cases of the Path-TSP: Garcia & Tejel [11] derive an $O(n \log n)$ algorithm for the Path-TSP with n cities on Convex-Euclidean distance matrices. The follow-up work [10] by Garcia, Jodra & Tejel uses sophisticated search techniques to improve the time complexity to linear time $O(n)$. Fig. 1 provides an example for the Convex-Euclidean Path-TSP, which is taken from Figure 3 in [11]; this example illustrates the diverse and manifold shapes of TSP-paths under Convex-Euclidean distances.

Now by looking deeper into the papers [10,11] and by carefully analysing the flow of arguments, one realizes that the approach does not exploit any geometric property of the Convex-Euclidean case that would go beyond the quadrangle inequalities (1)–(2). In other words, the arguments in [10,11] do not only settle the Path-TSP on Convex-Euclidean distance matrices, but they do also yield (without additional effort, and without changing a single letter) a polynomial time solution for the Path-TSP on Kalmanson matrices. Note that this step from Convex-Euclidean matrices to Kalmanson matrices for the Path-TSP runs perfectly in parallel with the step from Convex-Euclidean matrices to Kalmanson matrices for the classical TSP. Since the recognition problem for permuted Kalmanson matrices is solvable in polynomial time, the Path-TSP with a permuted Kalmanson matrix is also polynomially solvable (see the remark at the end of Section 1.1).



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
X	0	0	1	4	8	12	17	22	25	28	29	28	26	22	18	14	9	5	1
Y	6	10	11	13	15	15	15	14	13	10	6	5	3	1	0	0	0	1	3

Fig. 1. The Path-TSP in the Convex-Euclidean case: (a) An illustrating example from [11]; (b) An optimal (1,11)-path for the set of points listed in the table.

1.3. Contribution and organization of the paper

In this paper we take the logically next step in this line of research and show that the Path-TSP is polynomially solvable on Demidenko matrices. This substantially generalizes and extends the results in [10,11] for the Path-TSP on Convex-Euclidean matrices and Kalmanson matrices. We first analyse the combinatorial structure of an optimal TSP-path on Demidenko matrices, and prove that there always exists an optimal solution of a certain strongly restricted and nicely structured form. Then we show that we can optimize in polynomial time over the TSP-paths of that nicely structured form. By combining these results, we get our polynomial time result for Demidenko matrices. Notice, that the Demidenko matrices form a proper superset of the set of the Monge matrices, a class of matrices well known in the context of polynomially solvable special cases of hard combinatorial optimization problems, e.g. the TSP and the quadratic assignment problem. Thus, our result on the Path-TSP with a Demidenko distance matrix settles also the complexity of the Path-TSP with a Monge distance matrix.

The remainder of the paper is organized as follows. In Section 2 we summarize definitions and notations related to paths and tours as well as matrix classes of relevance in this paper. In Section 3 we introduce the concept of *forbidden pairs of arcs* and show that for arbitrary cities s and t there always exists an optimal (s, t) -TSP-path which does not contain forbidden pairs of arcs. Then we show some implications of this result in terms of the solution of the Path-TSP for $s = 1$ and $t = n$, and of structural properties of the optimal solution of the problem in the more general case with $s = 1$ and arbitrary t . Section 4 shows how to exploit the findings presented in Section 3 to efficiently solve the Path-TSP for $s = 1$ and for arbitrary t by dynamic programming. Finally, Section 5 shows how to efficiently solve the Path-TSP in the most general case, where there are no restrictions on s and t . Section 6 concludes the paper with some final remarks.

2. Definitions, notations and preliminaries

In this section we summarize various definitions and notations that will be used throughout the rest of the paper.

2.1. Paths, tours and related properties

We consider a set of n cities (sometimes also called *vertices*) with a symmetric $n \times n$ distance matrix $C = (c_{ij})$. We use the notation $\overline{a, b}$ for the set $\{a, a + 1, \dots, b\}$ of all integers between a and b , for any two integers a , and b with $a \leq b$.

A *tour* τ visiting k cities, $3 \leq k \leq n$, is a sequence $\tau = \langle \tau_1, \tau_2, \dots, \tau_k, \tau_{k+1} \rangle$ with $\tau_i \in \overline{1, n}$, $\tau_{k+1} = \tau_1$ and $\tau_1 < \tau_i$, for all $i \in \overline{2, k}$, such that every city from $\overline{1, n}$ except for τ_1 is contained at most once in the sequence. We say that the tour τ visits the cities τ_1, \dots, τ_k . A tour $\tau = \langle \tau_1 = 1, \tau_2, \dots, \tau_n, 1 \rangle$ visiting all n cities is called a *TSP tour*. An (s, t) -*path* visiting k cities is a sequence $\tau = \langle \tau_1 = s, \tau_2, \dots, \tau_k = t \rangle$ of cities which starts at s , ends at t , and contains every city from $\{\tau_1, \dots, \tau_k\} \subseteq \overline{1, n}$ exactly once, where $k \in \mathbb{N}$, $k \leq n$. We call s the start city (or vertex) and t the end city (or vertex). An (s, t) -*TSP-path* is an (s, t) -*path* visiting all n cities.

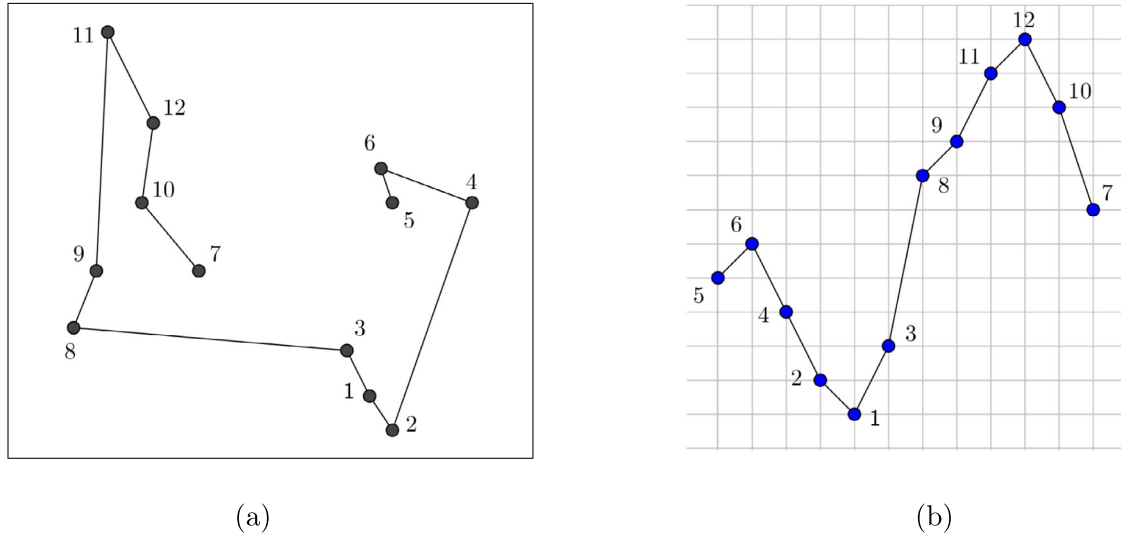


Fig. 2. (a) The optimal (5,7)-TSP-path for the set of points in Fig. 3; (b) Schematic representation of the path.

Given a tour or an (s, t) -path $\tau = \langle \tau_1, \tau_2, \dots, \tau_k \rangle$ and two indices $i \in \overline{1, k-1}, j \in \overline{2, k}$, then $\tau(\tau_i) := \tau_{i+1}$ is called the successor of τ_i in τ and $\tau^{-1}(\tau_j) := \tau_{j-1}$ is called the predecessor of τ_j in τ . An ordered pair $(\tau_i, \tau_{i+1} = \tau(\tau_i))$, for $i \in \overline{1, k-1}$, is called an *arc* in τ . If $\tau_i < \tau_{i+1}$ ($\tau_i > \tau_{i+1}$), (τ_i, τ_{i+1}) is called an *increasing arc* (*decreasing arc*). An (s, t) -path such that all its arcs are increasing (decreasing) arcs is called a *monotone increasing path* (*monotone decreasing path*). For $i \in \overline{2, k-1}$ a city τ_i reached along an increasing arc and left along a decreasing arc in τ is called a *peak*, in this case $\tau_i > \tau_{i-1}$ and $\tau_i > \tau_{i+1}$ hold. A city τ_i reached along a decreasing arc and left along an increasing arc in τ is called a *valley*, in this case $\tau_i < \tau_{i-1}$ and $\tau_i < \tau_{i+1}$ hold. The start city s of an (s, t) -path τ ($s \neq t$) is called a peak if it is left along a decreasing arc, i.e. if $s > \tau(s)$ holds. Analogously, s is called a valley if its left along an increasing arc, i.e. if $s < \tau(s)$ holds. Similarly, the end city t of an (s, t) -path τ ($s \neq t$) is called a peak (valley) if $\tau^{-1}(t) < t$ ($\tau^{-1}(t) > t$), i.e. if t is reached along an increasing (decreasing) arc. Given a tour or an (s, t) -path $\tau = \langle \tau_1, \tau_2, \dots, \tau_k \rangle$ and two indices $i, j \in \overline{1, k}, i \neq j$, the subsequence of τ starting at city τ_i and ending at city τ_j is called the (τ_i, τ_j) -subpath of τ .

A tour $\tau = \langle \tau_1, \tau_2, \dots, \tau_k, \tau_{k+1} \rangle$ is called *pyramidal* if it contains exactly one peak, or equivalently, if there exists an $l \in \overline{1, k}$ such that $\tau_1 < \dots < \tau_l$, and $\tau_l > \dots > \tau_k$ hold. A TSP tour which is pyramidal is called a *pyramidal TSP tour*.

An (s, t) -path $\tau = \langle \tau_1 = s, \tau_2, \dots, \tau_k = t \rangle$ is called λ -*pyramidal* if it contains exactly one peak, or equivalently there exists an index $l \in \overline{1, k}$ such that $\tau_1 < \tau_2 < \dots < \tau_l$ and $\tau_l > \tau_{l+1} > \dots > \tau_k$. Clearly, a path τ is λ -pyramidal iff it is the concatenation of a monotone increasing subpath with a monotone decreasing subpath in this order (where either one can be empty). An (s, t) -path $\tau = \langle \tau_1 = s, \tau_2, \dots, \tau_k = t \rangle$ is called ν -*pyramidal* if it contains exactly one valley, or equivalently, if there exists an index $l \in \overline{1, k}$ such that $\tau_1 > \tau_2 > \dots > \tau_l$ and $\tau_l < \tau_{l+1} < \dots < \tau_k$. In other words, a path τ is ν -pyramidal iff it is the concatenation of a monotone decreasing subpath with a monotone increasing subpath in this order (where either one can be empty). Notice that in the above definitions of λ -pyramidal paths and ν -pyramidal paths one of the chains of inequalities would become obsolete if $l = 1$ or $l = k$.

Finally we introduce a schematic representation of paths which is useful for the illustration of their combinatorial properties. An (s, t) -TSP-path $\tau = \langle \tau_1 = s, \dots, \tau_n = t \rangle, s < t$, visiting the n cities $\tau_i \in \overline{1, n}$ for $i \in \overline{1, n}$, is visualized on an $n \times n$ grid by placing city τ_i in the grid node with coordinates (i, τ_i) . For example, Fig. 2(a) shows the shortest (5, 7)-TSP-path $\tau, \tau = \langle 5, 6, 4, 2, 1, 3, 8, 9, 11, 12, 10, 7 \rangle$, for the set of 12 points in Fig. 3. The schematic representation of τ is shown in Fig. 2(b). Here the cities 6 and 12 are peaks and the cities 5, 1 and 7 are valleys. Thus, the path τ is neither λ -pyramidal nor ν -pyramidal. The $(1, 7)$ -subpath of τ is λ -pyramidal, whereas the $(6, 8)$ -subpath of τ is ν -pyramidal.

2.2. The computation of shortest pyramidal tours and paths

Next we introduce the quantities $E_m(i, j)$ and $D_w(i, j)$ for integers $i, j, m, w \in \overline{1, n}$, which represent the lengths of certain λ -pyramidal paths or ν -pyramidal paths, respectively. For $i, j, m \in \overline{1, n}$ with $i < j \leq m$ denote by $E_m(i, j)$ the length of a shortest λ -pyramidal (i, j) -path τ visiting the cities $\{i\} \cup \overline{j, m}$. Notice that due to the symmetry of the distance matrix $E_m(i, j)$ is also the length of a shortest λ -pyramidal (j, i) -path τ visiting the above set of cities. If $m = n$ we omit the subscript m in $E_m(i, j)$ and simply write $E(i, j)$. Clearly, in a λ -pyramidal (i, j) -path as above city $j + 1$ is either the successor of i or the predecessor of j . Analogously, in a pyramidal TSP-tour city 1 is visited right before or right after city 2. Thus, the optimal length of a pyramidal tour equals $E(1, 2) + c_{21}$. If $j = m$, then $E_m(i, m) = c_{im}$ and the corresponding path consists just of the arc (i, m) .

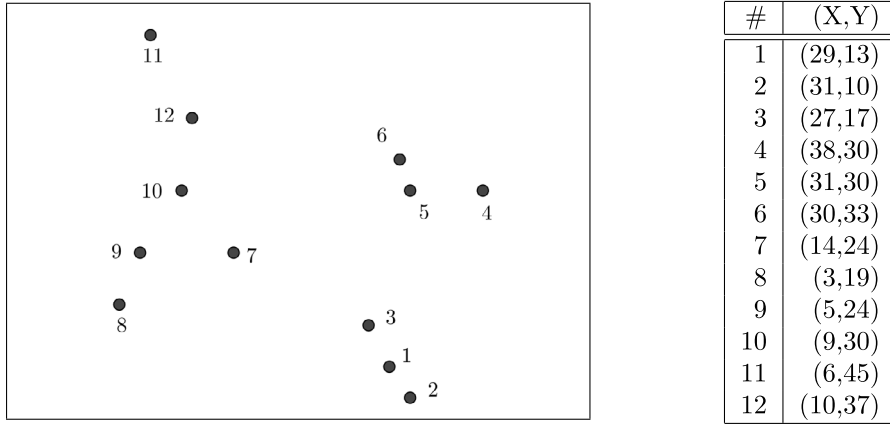


Fig. 3. The distance matrix of this set of points is a Demidenko matrix but not a Kalmanson matrix.

For $w, j, i \in \overline{1, n}$, $w \leq j < i$, let $D_w(i, j)$ be the length of a shortest v -pyramidal (i, j) -path which visits the cities $\overline{w, j} \cup \{i\}$ (and also the length of a shortest v -pyramidal (j, i) -path visiting the above set of cities). Clearly, in a v -pyramidal (i, j) -path as above city $j - 1$ is either the successor of i or the predecessor of j . If $j = w$, then $D_w(i, j) = c_{iw}$ and the corresponding path consists just of the arc (i, w) . If $w = q$ we omit the subscript w in $D_w(i, j)$ and simply write $D(i, j)$.

Observe that for every $m \in \overline{2, n}$, the quantities $E_m(i, j)$ with $i, j \in \overline{1, m}$, $i < j$, can be computed in $O(m^2)$ time by the dynamic programming recursions (3)–(4). Analogously for every $w \in \overline{1, n - 1}$, the quantities $D_w(i, j)$ with $j, i \in \overline{w, n}$, $j < i$, can be computed in $O((n - w)^2)$ time by the recursions (5)–(6).

$$E_m(i, j) = E_m(j, i) = \min\{E_m(j + 1, j) + c_{i,j+1}, E_m(i, j + 1) + c_{j+1,j}\},$$

for $i \in \overline{1, m - 2}, j \in \overline{i + 1, m - 1}$, and

$$E_m(i, m) = E_m(m, i) := c_{im}, \quad \text{for } i \in \overline{1, m - 1}. \tag{4}$$

$$D_w(i, j) = D_w(j, i) = \min\{D_w(j - 1, j) + c_{i,j-1}, D_w(i, j - 1) + c_{j-1,j}\},$$

for $j \in \overline{w + 1, n - 1}, i \in \overline{j + 1, n}$, and

$$D_w(i, w) = D_w(w, i) := c_{iw}, \quad \text{for } i \in \overline{w + 1, n}. \tag{6}$$

In particular the entries $E(i, j)$ and the entries $D(j, i)$, $i, j \in \overline{1, n}$, $i < j$, can be computed in $O(n^2)$ time.

Summarizing we get the following result:

Observation 2.1. *The quantities $E_m(i, j)$ for $i, j, m \in \overline{1, n}$, with $i < j \leq m$, can be computed in $O(n^3)$ time. Analogously the quantities $D_w(i, j)$ for $i, j, w \in \overline{1, n}$, with $w \leq j < i$, can be computed in $O(n^3)$ time. The quantities $E(i, j) := E_n(i, j)$ and $D(i, j) := D_1(j, i)$ with $i, j \in \overline{1, n}$, $i < j$ can be computed in $O(n^2)$ time.*

Since, as mentioned above, the optimal length of a pyramidal tour equals $E(1, 2) + c_{21}$ Observation 2.1 implies the following result known already in the 1970's.

Theorem 2.2 (Klyaus [18]; Gilmore, Lawler & Shmoys [13]). *A shortest pyramidal TSP tour visiting the cities $\overline{1, n}$ can be determined in $O(n^2)$ time.*

Let us now consider shortest λ -pyramidal paths and shortest v -pyramidal paths which visit *contiguous* sets of cities, respectively, i.e. sets of cities which consist of all cities k between i and j for some pair of cities $i, j \in \overline{1, n}$, $i < j$. For $i, p, m \in \overline{1, n}$, $i < p \leq m$, let $\Lambda_m(i, p)$ be the length of the shortest λ -pyramidal (i, p) -path which visits the cities $\overline{i, m}$. Due to the symmetry of the distance matrix $\Lambda_m(i, p)$ is also the length of a shortest λ -pyramidal (p, i) -path which visits the set of cities as above. In the special case $p = m$, $\Lambda_m(i, p)$ is the length of the monotone increasing path through the cities $\overline{i, m}$.

For $w, j, q \in \overline{1, n}$, $w \leq j < q$, let $V_w(j, q)$ be the length of a shortest v -pyramidal (j, q) -path visiting the cities $\overline{w, q}$ (and also the length of a shortest v -pyramidal (q, j) -path visiting the above set of cities). In the special case $w = j$, $V_w(j, q)$ is the length of the monotone increasing path through the cities $\overline{j, q}$.

If $m = n$ or $w = 1$ we omit the subscript m or w in $\Lambda_m(i, p)$ or $V_w(j, q)$ and write simply $\Lambda(i, p)$ or $V(j, q)$, respectively. The quantities $\Lambda_m(i, p)$, with $i, p, m \in \overline{1, n}$, $i < p \leq m$, and $V_w(j, q)$, with $w, j, q \in \overline{1, n}$, $w \leq j < q$, can also be computed efficiently by dynamic programming as shown in the following simple observation. We set $\Lambda_m(i, p) := 0$ or $V_w(j, q) := 0$, for $p = i = m$ or $w = j = q$, respectively.

Observation 2.3. The quantities $\Lambda_m(i, p)$ with $i, p, m \in \overline{1, n}$, $i < p \leq m$, and $V_w(j, q)$ with $w, j, q \in \overline{1, n}$, $w \leq j < q$, can be computed in $O(n^3)$ time. In particular, for $m = n$ and $w = 1$ the quantities $\Lambda(i, p) = \Lambda_n(i, p)$ with $i, p \in \overline{1, n}$, $i < p$, and $V(j, q) = V_1(j, q)$, with $j, q \in \overline{1, n}$, $j < q$, can be computed in $O(n^2)$ time.

Proof. For the quantities $\Lambda_m(i, p)$ the claim follows directly from [Observation 2.1](#) and the following equalities which hold for any triple (i, p, m) in the given range of indices:

$$\Lambda_m(i, p) = c_{i,i+1} + c_{i+1,i+2} + \dots + c_{p-2,p-1} + E_m(p-1, p), \text{ if } i < p-1, \text{ and}$$

$$\Lambda_m(i, p) = E_m(p-1, p), \text{ if } i = p-1.$$

Analogously, for the quantities $V_w(j, q)$ the claim follows directly from [Observation 2.1](#) and the following equalities which hold for any triple (w, j, q) in the corresponding range of indices:

$$V_w(j, q) = c_{q,q-1} + c_{q-1,q-2} + \dots + c_{j+2,j+1} + D_w(j+1, j), \text{ if } j < q-1, \text{ and}$$

$$V_w(j, q) = D_w(q, q-1), \text{ if } j = q-1.$$

Clearly the quantities $c_{i,i+1} + c_{i+1,i+2} + \dots + c_{p-2,p-1}$ can be computed in a preprocessing step in $O(n^3)$ time for all pairs (i, p) with $i, p \in \overline{1, n}$ and $i < p$. \square

In what follows we assume that all quantities $E_m(i, j)$, $D_w(i, j)$, $\Lambda_m(i, p)$, $V_w(j, q)$ with indices in their corresponding ranges are computed in a preprocessing step. Moreover we use the straightforward relationships

$$E_m(i, j) = \min \left\{ c_{i,k} + \Lambda_m(k, j) : k \in \overline{j+1, m} \right\} \text{ for } i < j, \text{ and}$$

$$D_w(i, j) = \min \left\{ c_{i,k} + V_w(j, k) : k \in \overline{w, j-1} \right\} \text{ for } i > j.$$

2.3. Classes of matrices

A symmetric $n \times n$ matrix $C = (c_{ij})$ is a *Kalmanson matrix*, if it satisfies the Kalmanson conditions in (1)–(2), and it is a *Demidenko matrix*, if it satisfies condition (1). In the later case (1) is also called the Demidenko condition. Note that a *principal submatrix* of an $n \times n$ Kalmanson (Demidenko) matrix C obtained from C by deleting the rows and columns with indices in a subset $S \subset \overline{1, n}$, $|S| \leq n-1$, is again a Kalmanson (Demidenko) matrix. The *reversed matrix* $D = (d_{ij})$ results from matrix C by simultaneously reversing the order of rows and columns in C ; in other words, the matrix is specified by $d_{ij} = c_{n+1-i, n+1-j}$, for $i, j \in \overline{1, n}$. A reversed Kalmanson (Demidenko) matrix is again a Kalmanson (Demidenko) matrix.

The example in [Fig. 3](#) shows that the Demidenko matrices form a proper superset of the Kalmanson matrices. It can be checked that the distance matrix C of the Euclidean distances of these 12 points in the Euclidean plane is a Demidenko matrix but not a Kalmanson matrix. Indeed, some points lie far from the boundary of the convex hull of all points and the matrix of their Euclidean distances is not even a permuted Kalmanson matrix. We refer the reader to Deineko, Rudolf, Van der Veen & Woeginger [7] for an explicit characterization of Euclidean sets of points that satisfy the Kalmanson conditions or the Demidenko conditions.

Back in 1979, Demidenko [9] proved that an optimal TSP tour on a set of cities with a Demidenko distance matrix can be found among the pyramidal TSP-tours. Together with [Theorem 2.2](#) this implies the following result:

Theorem 2.4 (Demidenko [9]). *For an n -city TSP instance with a Demidenko distance matrix, an optimal TSP tour can be found among the pyramidal TSP tours; hence it can be determined in $O(n^2)$ time.*

Throughout this paper, we will assume that all considered distance matrices have non-negative entries. This assumption can be made without loss of generality, as Demidenko matrices can be transformed into non-negative Demidenko matrices by simply adding a sufficiently large constant to each entry. Clearly the Path-TSP with a distance matrix $C = (c_{ij})$ and the Path-TSP with a distance matrix $\tilde{C} = (c_{ij} + K)$ for some $K \in \mathbb{R}$ are equivalent, in the sense that the sets of their optimal solutions coincide.

3. Forbidden pairs of arcs and structural properties of optimal TSP-paths starting at city 1

In this section we investigate the combinatorial structure of optimal (s, t) -TSP-paths in the case where the distance matrix of the cities is a Demidenko matrix. An essential concept used in our investigations is that of a *forbidden pair of arcs*. We show first that there always exists an optimal (s, t) -TSP-path which does not contain forbidden pairs of arcs. As implications of this fact we obtain the solution of the $(1, n)$ -Path-TSP and further structural properties of the optimal $(1, t)$ -TSP-path for $t \neq 1$.

Definition 3.1. In an (s, t) -path τ a pair of arcs $(i, \tau(i))$ and $(j, \tau(j))$ is called a *forbidden pair of arcs* if either $i < j < \tau(i) < \tau(j)$ or $i > j > \tau(i) > \tau(j)$ holds.

Lemma 3.2. Consider a Path-TSP instance with a Demidenko distance matrix (c_{ij}) . There exists an optimal (s, t) -TSP-path which does not contain forbidden pairs of arcs.

Proof. Let $\tau = \langle \tau_1 = s, \tau_2, \dots, \tau_n = t \rangle$ be an optimal (i.e. shortest) (s, t) -TSP-path with some forbidden pair of arcs $(i, \tau(i))$ and $(j, \tau(j))$. Assume without loss of generality that $i < j < \tau(i) < \tau(j)$ and that city i is reached earlier than city j in τ . Apply a standard transformation technique (see for instance Burkard & al [3]) to construct an optimal (s, t) -TSP-path which does not contain the pair $(i, \tau(i))$ and $(j, \tau(j))$ of forbidden arcs: invert the $(\tau(i), j)$ -subpath of τ into $(j, \dots, \tau(i))$, and replace the forbidden pair of arcs by the new pair of arcs (i, j) and $(\tau(i), \tau(j))$. Clearly the resulting path τ' is an (s, t) -TSP-path. Moreover the inequalities (1) imply that $c_{i\tau(i)} + c_{j\tau(j)} \geq c_{ij} + c_{\tau(i)\tau(j)}$, and therefore the length of τ' does not exceed the length of τ . So τ' is an optimal (s, t) -TSP-path which does not contain the pair $(i, \tau(i))$ and $(j, \tau(j))$ of forbidden arcs. If this path still contains a forbidden pair of arcs we apply the above transformation again and repeat this process as long as the current optimal (s, t) -TSP-path contains a forbidden pair of arcs.

In order to see that this transformation process terminates after a final number of steps consider a potential function K which maps any (s, t) -TSP-path π to a non-negative integer $K(\pi) := \sum_{i=1, i \neq t}^n |i - \pi(i)|$. It can easily be seen that the transformation described above reduces the value of the potential function, i.e. $K(\tau') < K(\tau)$. Since the potential function takes only non-negative integer values the process stops after a final number of steps. \square

Lemma 3.2 implies the following result on the $(1, n)$ -TSP-paths.

Theorem 3.3. $\langle 1, 2, \dots, n \rangle$ is a shortest $(1, n)$ -TSP-path for the Path-TSP with a Demidenko distance matrix.

Proof. We show that any $(1, n)$ -TSP-path with a non-trivial peak, i.e. a peak different from n , contains a forbidden pair of arcs. The proof of the lemma is then completed by observing that $\langle 1, 2, \dots, n \rangle$ is the unique $(1, n)$ -TSP-path without a non-trivial peak.

Let m be the first peak in a $(1, n)$ -TSP-path τ . Since m is a peak, there is an arc (i, j) in a (m, n) -subpath of τ such that $i < m < j$. If $\tau^{-1}(m) < i$, then $(\tau^{-1}(m), m)$ and (i, j) build a forbidden pair of arcs. If $\tau^{-1}(m) > i$, then on the monotone increasing $(1, m)$ -subpath of τ there exists an arc (k, l) such that $k < i < l \leq \tau^{-1}(m)$. By observing that $\tau^{-1}(m) < m < j$ we conclude that the pair (k, l) and (i, j) is a forbidden pair of arcs in this case. \square

The following corollary is a statement about the monotonicity of peaks and valleys in optimal $(1, t)$ -TSP-paths.

Corollary 3.4. Consider a Path-TSP on n cities with a Demidenko distance matrix. For any $t \in \overline{1, n}$, $t \neq 1$, there exists an optimal $(1, t)$ -TSP-path with peaks decreasing and valleys increasing from the left to the right in the path, i.e. if peak p (valley v) is reached earlier than peak p' (valley v') in the path, then $p > p'$ ($v < v'$) holds.

Proof. The proof is done by induction on the number n of cities. The statement is trivially true for $n = 2$. So assume that $n > 2$.

The correctness of the statement for $t = n$ follows immediately from Theorem 3.3: the $(1, n)$ -TSP-path $\langle 1, 2, \dots, n \rangle$ contains just the (trivial) valley 1 and the (trivial) peak n . Thus we assume without loss of generality that $t \neq n$ and let τ be an optimal $(1, t)$ -TSP-path. Let $\tau' = \langle \tau_1 = 1, \tau_2, \dots, \tau_k = n \rangle$, with $k \in \overline{1, n}$, $k < n$, be the $(1, n)$ -subpath of τ . Since a principal submatrix of a Demidenko matrix is a Demidenko matrix, as mentioned in Section 2.3, by applying Theorem 3.3 we can reorder the cities of τ' increasingly and obtain a $(1, n)$ -path of the same length which visits the same cities as τ' . So we can assume without loss of generality that there is no peak in the $(1, n)$ -subpath τ' but n , and hence n is also the first peak in $(1, t)$ -TSP-Path τ . Now we distinguish two cases: (a) the last city t is the smallest city in the (n, t) -subpath of τ and (b) there is a city j with $j < t$ in the (n, t) -subpath of τ . In Case (a) we can assume without loss of generality that there are no other peaks but n in the (n, t) -subpath of τ (by applying similar arguments to the one mentioned above for the $(1, n)$ -subpath τ' of τ); this assumption is justified by Theorem 3.3 and by the fact that a reversed Demidenko matrix is a Demidenko matrix. So, we assume w.l.o.g. that the (n, t) -subpath of τ is monotone decreasing the statement of the corollary holds in this case. (Notice, that in this case we can find an optimal $(1, t)$ -TSP-path which is λ -pyramidal.)

In Case (b) the (n, t) -subpath from n to t contains at least one valley which is smaller than t . Let $v = \tau_l < t$ be the smallest valley in the (n, t) -subpath of τ , for $k < l < n$. Analogously as for the $(1, n)$ -subpath and for Case (a) we can assume without loss of generality that the (n, v) -subpath of τ contains no peaks, but n . Denote by $\bar{\tau}$ the (v, t) -subpath of τ , $1 < v < t < n$. Since the distance matrix of the cities visited by $\bar{\tau}$ is a Demidenko matrix (as a principal submatrix of a Demidenko matrix), the induction hypothesis applies and this completes the proof. \square

Lemma 3.5. Consider a Path-TSP with a Demidenko distance matrix and an optimal $(1, t)$ -TSP-path τ which contains no forbidden pairs of arcs and such that its peaks decrease and its valleys increase from the left to the right in the path. Let m_1 and m_2 , $m_1 > m_2$, be two consecutive peaks in τ . Let w_1 be the valley that precedes peak m_1 , let w_2 be the valley that follows m_1 and precedes m_2 , and let w_3 be the valley that follows m_2 . Then the following statements hold:

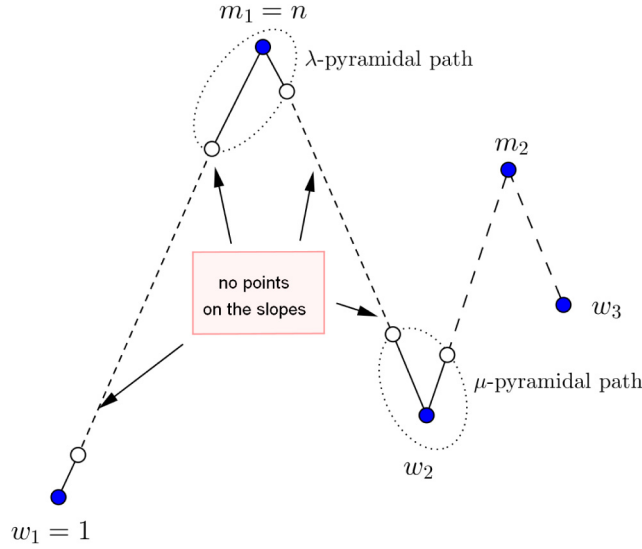


Fig. 4. An illustration to the structure of a path without forbidden arcs.

- (i) The (w_1, m_1) -subpath of τ contains no city i for which $w_2 < i < m_2$ holds.
- (ii) The (m_1, w_2) -subpath of τ contains no city j for which $w_3 < j < m_2$ holds.

Proof. Fig. 4 illustrates the structure of a path with the properties described in the lemma. We prove here only statement (i), statement (ii) can be proved by using similar arguments.

Consider the (w_1, m_1) -subpath of τ and assume that (i) does not hold. Let i be the largest city on the (w_1, m_1) -subpath such that $w_2 < i < m_2$. If $i > \tau^{-1}(m_2)$, then $\tau^{-1}(m_2) < i < m_2 < \tau(i)$ and hence $(i, \tau(i)), (\tau^{-1}(m_2), m_2)$ build a forbidden pair of arcs contradicting the assumption of the lemma. If $i < \tau^{-1}(m_2)$, then on the $(w_2, \tau^{-1}(m_2))$ -subpath of τ there exists an arc (k, l) such that $k < i < l$. In this case $(i, \tau(i))$ and (k, l) build a forbidden pair of arcs and this contradicts the assumption of the lemma. \square

In particular, the statements in the above lemma imply that the cities $\overline{m_2 + 1, m_1}$ are placed on consecutive positions in path τ and form a λ -pyramidal subpath of it.

4. Efficient solution of the Path-TSP with a Demidenko distance matrix: the case $s = 1$

In this section we exploit the structural properties of optimal $(1, t)$ -TSP-paths identified in Section 3 and derive recursive equations for the length $H_n(1, t)$ of an optimal $(1, t)$ -TSP-path through the cities $\overline{1, n}$. These equations yield then an $O(n^2)$ dynamic programming algorithm for the solution of the Path-TSP with a Demidenko distance matrix where the starting city is 1 and the destination city arbitrary.

The recursive equations for $H_n(1, t)$. Due to Lemma 3.2, Corollary 3.4 and Lemma 3.5, we consider without loss of generality an optimal $(1, t)$ -TSP-paths τ which contains no forbidden pairs of arcs, has decreasing peaks and increasing valleys from the left to the right and fulfills the statements of Lemma 3.5. We distinguish two cases: (1) the optimal $(1, t)$ -TSP-path contains no valleys but 1 and t , and (2) the optimal $(1, t)$ -TSP-path contains at least one valley w with $1 < w < t$. In the first case the optimal $(1, t)$ -TSP-path is λ -pyramidal and thus $H_n(1, t) = \Lambda_n(1, t)$. In the second case let $w = j + 1, j \geq 1, w < t$ be the left-most (and the smallest) non-trivial valley in τ .

Since n is the first peak, valley w is reached after n in τ . According to Lemma 3.5 (see also Fig. 4 where the role of w is played by w_2), τ starts with a monotone increasing $(1, w - 1)$ -subpath visiting the cities $\overline{1, j}$, followed first by a λ -pyramidal path with the peak $m = n$, then by a μ -pyramidal path with valley $w = j + 1$, and so on, until the final city t is reached. For cities j, m, w such that $j < w < m$ and $w \leq t < m$ denote by $\Gamma(j, m, w)$ the length of an optimal (j, t) -path starting at j and then visiting the cities of the set $\{j\} \cup \overline{w, m}$ with the first peak in this path being m , and the valley that follows m being $w > j$. Here w could also coincide with t , in which case t would be reached along a decreasing sequence of cities from m to t . Then the length $H_n(1, t)$ of the optimal $(1, t)$ -TSP-path is given as follows

$$H_n(1, t) = \min\{\Lambda_n(1, t), \min\{V_1(1, j) + \Gamma(j, n, j + 1) : j = \overline{1, t - 2}\}\}. \quad (7)$$

where $V_1(1, j)$ is defined as in Section 2.1.

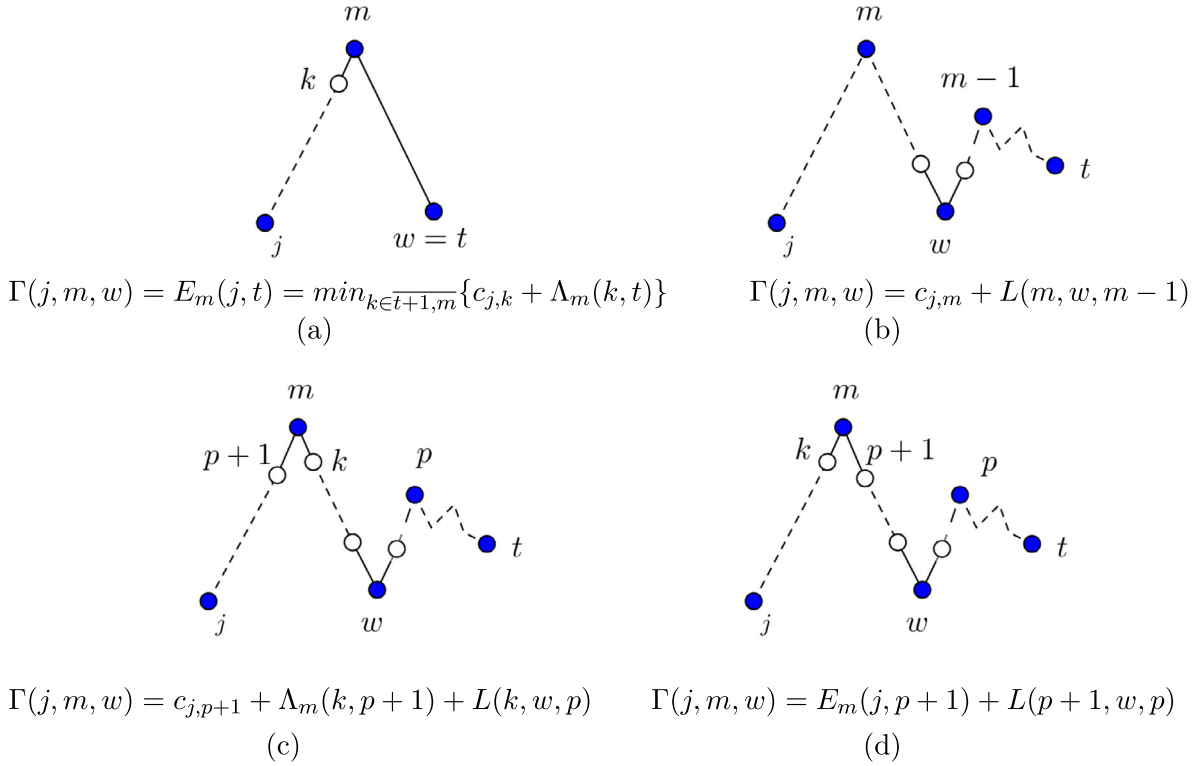


Fig. 5. An illustration to the calculation of $\Gamma(j, m, w)$. The straight dashed lines represent single edges.

Next we give a recursive equation for the computation of the quantities Γ above. To this end denote by $L(k, w, p)$ the length of an optimal (k, t) -path visiting the cities of the set $\overline{w, p} \cup \{k\}$ with first valley w and first peak p such that w precedes p , for cities w, p, k such that $w < p < k$ and $w < t \leq p$. Here p could also coincide with t in which case t would be reached along an increasing sequence of cities from w to t .

According to Lemma 3.2, Corollary 3.4 and Lemma 3.5 and as illustrated in Fig. 5 the values $\Gamma(j, m, w)$, for $j, m, w \in \overline{1, n}$ with $j < w \leq t < m$, can be calculated by the following recursions:

$$\Gamma(j, m, w) = E_m(j, t), \text{ if } w = t, \text{ and}$$

$$\Gamma(j, m, w) = \min \left\{ c_{j,m} + L(m, w, m-1), \min_{\{ \alpha_{j,m,w}(p, k) : p \in \overline{t, m-2}, k \in \overline{p+2, m} \}} \right\}, \text{ if } w < t, \quad (8)$$

where $\alpha_{j,m,w}(p, k) := \min\{c_{j,p+1} + \Lambda_m(k, p+1) + L(k, w, p), E_m(j, p+1) + L(p+1, w, p)\}$.

The above recursions (8) cover the following cases which are also illustrated in Fig. 5.

Case $w = t$. The (j, w) -path has only one peak m (since $w = t$), starts at j , ends at t and goes through the cities $\overline{t+1, m}$. The optimal length of the pyramidal path on these cities equals $E_m(j, t)$ (see the definition before Observation 2.1). This case corresponds to the first line in (8) and is illustrated in Fig. 5(a).

Case $w \neq t$. In this case there will be another peak (different from m) to the right of the valley w . Let this peak be p , with $p \leq m-1$ according to Corollary 3.4. We again distinguish two subcases: $p = m-1$ and $p < m-1$. If $p = m-1$, then according to Lemma 3.5 the path starts with the arc (j, m) followed by (m, t) -subpath which starts at m and goes through the cities $\overline{w, m-1}$, with the first valley in this path being w and peak $m-1$ reached after valley w . The optimal length of such a path is given by $L(m, w, m-1)$ and in this case $\Gamma(j, m, w)$ is calculated as shown in the second line of (8) and illustrated in Fig. 5(b).

If $p < m-1$, then the (j, t) -path starts with an arc connecting j to the left-most city of a λ -pyramidal path with peak m containing the cities $\overline{p+1, m}$ and either starting or ending at city $p+1$. Let the other end-city of this λ -pyramidal path be $k \in \overline{p+2, m}$. Then the length of the (j, t) -path is calculated as shown in the third line of (8) and illustrated in Figs. 5(c) and 5(d). These pictures correspond to the cases where above mentioned λ -pyramidal path starts or ends at $p+1$, respectively.

The values $L(k, w, p)$, for $w, k, p \in \overline{1, n}$ with $1 < w < t \leq p < k \leq n$, can be computed recursively in a similar way:

$$L(k, w, p) = D_w(k, t), \text{ if } p = t, \text{ and}$$

$$L(k, w, p) = \min \begin{cases} \min\{\beta_{k,w,p}(v, j) : v \in \overline{w+2, t}, j \in \overline{w, v-2}\} \\ c_{k,w} + \Gamma(w, p, w+1), \end{cases} \text{ if } p > t, \quad (9)$$

where $\beta_{k,w,p}(v, j) := \min\{D_w(k, v-1) + \Gamma(v-1, p, v), c_{k,v-1} + V_w(j, v-1) + \Gamma(j, p, v)\}$.

Based on Eqs. (7)–(9) we obtain the following result about the computation of the optimal $(1, t)$ -TSP path in the case of a Demidenko distance matrix.

Theorem 4.1. Consider a Path-TSP on n cities $\overline{1, n}$ with a Demidenko distance matrix. An optimal $(1, t)$ -TSP-path can be found in $O(n^5)$ time.

Proof. Recall that all values $\Lambda_m(i, p)$, for $i, p, m \in \overline{1, n}$ with $i < p \leq m$, and $V_w(j, q)$, for $w, j, q \in \overline{1, n}$, with $w \leq j < q$, can be calculated in $O(n^3)$ time in a preprocessing step, see [Observation 2.3](#). Further, according to Eq. (7) the computation of the length $H_n(1, t)$ of the optimal $(1, t)$ -TSP-path (for $t \geq 3$) involves the quantities $\Gamma(j, n, j+1)$, for $j \in \overline{1, t-2}$. The quantities $\Gamma(j, m, w)$, for $j, w, m \in \overline{1, n}$ with $j < w \leq t < m$, are computed recursively together with the quantities $L(k, w, p)$, for $w, p, k \in \overline{2, n}$ with $w < t \leq p < k$. Observe that the computation of $\Gamma(j, m, w)$, for some (j, m, w) in the corresponding range, just involves quantities $L(x, y, z)$ for which the difference $z - y$ between the specified peak z and valley y is strictly smaller than the difference $m - w$ between the specified peak m and the specified valley w in $\Gamma(j, m, w)$. Analogously, the computation of $L(k, w, p)$, for some (k, w, p) in the corresponding range, involves quantities $\Gamma(x, y, z)$ for which the difference $y - z$ between the specified peak y and valley z is strictly smaller than the difference $p - w$ between the specified peak p and the specified valley w in $L(k, w, p)$. So the recursion would start with the trivial values $\Gamma(j, t+1, t) := c_{j,t+1} + c_{t+1,t}$ for $j, t \in \overline{1, n}$ with $j < t$, and $L(k, t-1, t) := c_{k,t-1} + c_{t-1,t}$ for $k, t \in \overline{1, n}$ with $k > t$. The effort needed for the recursive computation of all other values of Γ and L is the computation of the minimum over $O(n^2)$ sums consisting of previously computed entries of the same arrays L, Γ and of appropriate entries of the arrays E, Λ, D, V all of which have been computed in a preprocessing step. Hence the computation of each of the $O(n^3)$ values $\Gamma (L)$ can be done in $O(n^2)$ time and the overall time complexity is $O(n^5)$. \square

5. Efficient solution of the Path-TSP with a Demidenko distance matrix: the general case

This section deals with the polynomial time computation of an optimal (s, t) -TSP-path in the most general case where $s, t \in \overline{1, n}$, and none of the conditions $s = 1$ and $t = n$ is necessarily fulfilled. The results are stated in [Theorem 5.4](#) and the rest of the section is dedicated to its proof.

The following assumptions hold throughout the rest of this section. Due to the symmetry of the distance matrix we can assume that $s < t$ holds. Moreover, we assume that $s > 1$ because the case $s = 1$ has already been handled in [Theorem 4.1](#). The case $t = n$ can be handled analogously to the case $s = 1$, since a reversed Demidenko matrix is again a Demidenko matrix, as pointed out in Section 2. Summarizing we assume without loss of generality that $1 < s < t < n$ holds. Further, by [Lemma 3.2](#) we only consider (s, t) -TSP-paths without forbidden pairs of arcs. The following three claims state some particular structural properties of such paths which will be useful for the proof of [Theorem 4.1](#).

Claim 5.1. There is an optimal (s, t) -TSP-path τ with $1 < s < t < n$, where city 1 precedes city n .

Proof. To prove the claim we show that any (s, t) -TSP-path τ in which n precedes 1 contains a forbidden pair of arcs. Indeed, the (s, n) -subpath of τ contains an arc (u, v) with $u < p < v$, where p is the first peak in the subpath from 1 to t . (If the subpath from 1 to t is monotone we have $p = t$.) If $\tau^{-1}(p) < u$, then the arcs (u, v) and $(\tau^{-1}(p), p)$ build a forbidden pair of arcs. Otherwise consider an arc (x, y) in the (monotone) subpath from 1 to p such that $x < u < y$. The arcs $(u, v), (x, y)$ build a forbidden pair of arcs in this case. \square

Claim 5.2. There is an optimal (s, t) -TSP-path in which 1 precedes n and each city of the $(s, 1)$ -subpath is smaller than each city of the (n, t) -subpath.

Proof. To prove the claim we consider an arbitrary (s, t) -TSP-path τ in which city 1 precedes city n and show that the existence of a city in the $(s, 1)$ -subpath which is larger than some city in the (n, t) -subpath implies the existence of a forbidden pair of arcs. In particular if i is a city in the $(s, 1)$ -subpath of τ and j is a city in the (n, t) -subpath of τ such that $i > j$, it can be shown by arguments similar to those in the proof of [Claim 5.1](#) that the (i, j) -subpath of τ contains a forbidden pair of arcs. \square

Claim 5.3. There exists an optimal (s, t) -TSP-path τ which is a concatenation of two paths τ_1^p and τ_2^p such that τ_1^p starts at s and visits all cities from the set $\overline{1, p-1}$, and τ_2^p starts at the last city of τ_1^p , then visits all cities from the set $\overline{p, n}$ and ends at t , for some $p \in \overline{s+1, t}$.

Proof. To prove the claim we consider an (s, t) -TSP-path τ in which city 1 precedes city n and such that each city of the $(s, 1)$ -subpath of τ is smaller than each city of the (n, t) -subpath of τ . Claim 5.2 guarantees the existence of such a path. Observe that due to Theorem 3.3 and the fact that a principal submatrix of a Demidenko matrix is a Demidenko matrix, the $(1, n)$ -subpath of τ is monotone increasing. Now let p be the smallest city on the (n, t) -subpath of τ and let x be the last city with $x < p$, $x \neq s$, in the $(1, n)$ -subpath of τ . Then the cities in $\bar{1}, p - 1$ are exactly the cities visited by the (s, x) -subpath of τ . Hence we can set τ_1^p and τ_2^p as the (s, x) -subpath and the (x, t) -subpath of τ , respectively. \square

Theorem 5.4. Consider a Path-TSP on n cities $\bar{1}, n$ with a Demidenko distance matrix and a given pair of cities (s, t) . An optimal (s, t) -TSP-path can be found in $O(|t - s|n^5)$ time.

Proof. Due to Claim 5.1–5.3 we minimize over (s, t) -TSP-paths τ in which city 1 precedes city n , each city in the $(s, 1)$ -subpath has a smaller index than each city in the (n, t) -subpath, and for which an index $p \in s + 1, t$ exists such that τ is the concatenation of τ_1^p and τ_2^p as described in Claim 5.3. We refer to τ_1^p and τ_2^p as the prefix and the postfix of τ and denote by x the last city of the prefix (and the first city of the postfix). Next we show how to efficiently determine the length of a shortest prefix and a shortest postfix for each $p \in s + 1, t$ and each $x, x < p, x \neq s$.

By using the fact that a principal submatrix of a Demidenko matrix is a Demidenko matrix and by considering that the postfix starts at $x, x < p$, and then visits all cities in $\{x\} \cup \bar{p}, n$, the shortest length $T(x, p)$ of the postfix can be determined in $O(n^5)$ as described in Theorem 4.1, for every $p \in s + 1, t$ and for every $x < p$. Now let us virtually shrink the postfix to its second city $\tau(x)$. To distinguish $\tau(x)$ from the shrunk city let us denote the later by ν_x^p . We set the distance between ν_x^p and x equal to $T(x, p)$. Observe that the index of city $\tau(x)$ is larger than the indices of all cities in the prefix (with indices lying in $\bar{1}, p - 1$), hence the dummy city ν_x^p can be considered to have index p . Consider now the path $\bar{\tau}_1^p$ obtained by extending the prefix along the edge (x, ν_x^p) ; since τ contains no forbidden pairs of arcs also $\bar{\tau}_1^p$ contains no forbidden pairs of arcs. Notice that a shortest (t, n) -TSP-path without a forbidden pair of arcs can be determined in the same way as a shortest $(1, t)$ -TSP-path without a forbidden pair of arcs (after renumbering the rows and columns of the distance matrix from the right to the left prior to the calculations). Thus the length $\bar{T}(x, p)$ of $\bar{\tau}_1^p$ which starts at s , visits all cities from $\bar{1}, p - 1$ and then ends at ν_x^p which has index p , can be computed in $O(n^5)$ time as described in Theorem 4.1 (recall that the proof of Theorem 4.1 relies exclusively on the properties of paths without forbidden pairs of arcs). Clearly, $\bar{T}(x, p)$ equals the sum of the lengths of the prefix and the postfix, given $p \in s + 1, t$ and $x < p, x \neq s$. Consequently, the required length of the shortest (s, t) -TSP-path equals $\min\{\bar{T}(x, p) : p \in s + 1, t, x < p, x \neq s\}$. It is straightforward to compute this minimum in $O((t - s)n^6)$ time after having computed $\bar{T}(x, p)$ in $O(n^5)$ time for each pair of indices x and p as above. A closer look at the recursions (7) and Eqs. (8)–(9) involved in the computation of $\bar{T}(x, p)$ and $T(x, p)$ according to Theorem 4.1, reveals that for each fixed p all computations use the same quantities Γ and L , independently on the value of x , where $x < p$ and $x \neq s$. Thus for any $p \in s + 1, t$ all values of $\bar{T}(x, p)$ (and also $T(x, p)$) for $x < p, x \neq s$, can be computed in $O(n^5)$ time leading to an overall time complexity of $O((t - s)n^5)$ and completing the proof. \square

6. Final remarks

We have analysed the Path-TSP with a Demidenko distance matrix and have derived a sophisticated algorithm with a polynomial time complexity of $O(n^6)$ for this problem. An obvious open challenge is to get an improvement to some more civilized time complexity like $O(n^2)$, or at least $O(n^3)$.

The polynomially solvable special case for Demidenko distance matrices yields an exponential neighbourhood over which we can optimize in polynomial time. Such exponential neighbourhoods, also called very large-scale neighbourhoods (VLSN) could be used in local search approaches for the Path-TSP. Indeed, there has been quite some research on VLSN for different NP-hard combinatorial optimization problems and, in particular, also for the TSP. Substantial results concerning the construction of VLSN for the classical TSP, their combinatorial properties and the theoretical complexity aspects of optimizing over such neighbourhoods can be found in Ahuja, Ergun, Orlin & Punnen [1], Deineko & Woeginger [8], Gutin, Yeo & Zverovich [15], and Orlin & Sharma [21]. Among other VLSNs the above authors mention the pyramidal tours neighbourhood (PTN) which seems to be the simplest VLSN induced by a polynomial-time solvable special case of the TSP and is defined as follows. For a given permutation ϕ of $\{1, 2, \dots, n\}$ the corresponding neighbourhood $PTN(\phi)$ contains all permutations obtained as $\phi \circ \pi$, where π is any pyramidal tour of the cities $\{1, 2, \dots, n\}$. The construction of efficient local search algorithms for the TSP (and also for the Path-TSP) based on PTN and the analysis of their performance is a topic open for further research. In particular questions related to the connectivity and the diameter of the PTN graph, defined below, are of both theoretical and practical interest in this context. The vertices of the PTN graph represent the permutations of the n cities, i.e. the solutions of the TSP. There is a directed edge from permutation ϕ to permutation ψ iff $\psi \in PTN(\phi)$, i.e. iff ψ is in the PTN of ϕ . Other interesting open questions are the dominance analysis and the complexity of local search with respect to PTN. From a practical point of view the efficient optimization over the PTN seems to be challenging, despite the moderate time complexity ($O(n^2)$) needed to optimize over $PTN(\phi)$ for some arbitrary permutation ϕ . A viable and more efficient alternative to the optimization over $PTN(\phi)$ could be the search for approximate local optima in each local search iteration, in the vein of Orlin, Punnen and Schulz [20]. In the context of local search approaches to solve the Path-TSP the questions mentioned above are also in place for the VLNS arising from the new polynomially solvable special case discussed in this paper.

A different question open to further research concerns specific polynomially solvable special cases of the metric Path-TSP. Imposing the triangle inequality on the entries of the distance matrix essentially changes the computational complexity of the problem, see [25]. Thus, new polynomially solvable special cases, involving less restrictive four-points conditions, could arise for the metric Path-TSP.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was conducted while Vladimir Deineko and Gerhard Woeginger were visiting TU Graz, and they both thank the Austrian Science Fund (FWF): W1230, Doctoral Program in “Discrete Mathematics” for the financial support. Vladimir Deineko acknowledges support by Warwick University’s Centre for Discrete Mathematics and Its Applications (DIMAP). Gerhard Woeginger acknowledges support by the DFG RTG 2236 “UnRAVeL” – Uncertainty and Randomness in Algorithms, Verification and Logic.

References

- [1] R.K. Ahuja, O. Ergun, J.B. Orlin, A.P. Punnen, A survey of very large-scale neighborhood search techniques, *Discrete Appl. Math.* 123 (2002) 75–102.
- [2] D.L. Applegate, R.E. Bixby, V. Chvátal, W.J. Cook, *The Traveling Salesman Problem: A Computational Study*, Princeton University Press, 2006.
- [3] R.E. Burkard, V.G. Deineko, R. van Dal, J.A.A. van der Veen, G.J. Woeginger, Well-solvable special cases of the TSP: A survey, *SIAM Rev.* 40 (1998) 496–546.
- [4] E. Çela, V. Deineko, G.J. Woeginger, On x-and-y axes travelling salesman problem, *European J. Oper. Res.* 223 (2012) 333–345.
- [5] W.J. Cook, *In Pursuit of the Travelling Salesman: Mathematics and the Limits of Computation*, Princeton University Press, 2012.
- [6] V.G. Deineko, B. Klinz, A. Tiskin, G.J. Woeginger, Four-point conditions for the TSP: The complete classification, *Discrete Optim.* 14 (2014) 147–159.
- [7] V.G. Deineko, R. Rudolf, J.A.A. Van der Veen, G.J. Woeginger, Three easy special cases of the Euclidean traveling salesman problem, *RAIRO Oper. Res.* 31 (4) (1997) 342–362.
- [8] V.G. Deineko, G.J. Woeginger, A study of exponential neighborhoods for the travelling salesman problem and for the quadratic assignment problem, *Math. Program.* 87 (2000) 519–542.
- [9] V.M. Demidenko, The travelling salesman problem with asymmetric matrices, *Vestsi Akad. Navuk BSSR Ser. Fiz.-Mat. Navuk* 1 (1979) 29–35, (in Russian).
- [10] A. Garcia, P. Jodra, J. Tejel, An efficient algorithm for on-line searching in Monge path-decomposable tridimensional arrays, *Inform. Process. Lett.* 68 (1998) 3–9.
- [11] A. Garcia, J. Tejel, Using total monotonicity for two optimization problems on the plane, *Inform. Process. Lett.* 60 (1996) 13–17.
- [12] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide To the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
- [13] P.C. Gilmore, E.L. Lawler, D.B. Shmoys, Well-solved special cases, in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (Eds.), Chapter 4 in the *Traveling Salesman Problem*, Wiley, Chichester, 1985, pp. 87–143.
- [14] G. Gutin, A.P. Punnen, *The Travelling Salesman Problem and Its Variations*, Kluwer Academic Publishers, 2002.
- [15] G. Gutin, A. Yeo, A. Zverovich, Exponential neighborhoods and domination analysis for the TSP, in: G. Gutin, A.P. Punnen (Eds.), Chapter 6 in the *Travelling Salesman Problem and Its Variations*, Kluwer Academic Publishers, 2002, pp. 223–256.
- [16] S.N. Kabadi, Polynomially solvable cases of the TSP, in: G. Gutin, A.P. Punnen (Eds.), Chapter 11 in the *Travelling Salesman Problem and Its Variations* (2002) 489–583.
- [17] K. Kalmanson, Edgeconvex circuits and the travelling salesman problem, *Canad. J. Math.* 27 (1975) 1000–1010.
- [18] P.S. Klyaus, The structure of the optimal solutions of some classes of the traveling salesman problem, *Izv. Akad. Nauk. BSSR, Ser. Fiz.-Mat. Nauk* 6 (1976) 95–98, (in Russian).
- [19] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, *The Traveling Salesman Problem*, Wiley, Chichester, 1985.
- [20] J.B. Orlin, A.B. Punnen, A.S. Schulz, Approximate local search in combinatorial optimization, *SIAM J. Comput.* 33 (5) (2003) 1201–1214.
- [21] J.B. Orlin, D. Sharma, The extended neighborhood: Definition and characterization, *Math. Program.* 101 (2004) 537–559.
- [22] C.H. Papadimitriou, M. Yannakakis, The travelling salesman problem with distances one and two, *Math. Oper. Res.* 18 (1993) 1–11.
- [23] V. Traub, J. Vygen, R. Zenklusen, Reducing Path TSP to TSP, in: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, 2020*, pp. 14–27, To appear in *SIAM Journal on Computing*. (see also [arXiv:arXiv:1907.10376\[cs.DM\]](https://arxiv.org/abs/1907.10376)).
- [24] D.P. Williamson, D.B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, New York, 2011.
- [25] R. Zenklusen, A 1.5-Approximation for path TSP, in: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA-2019, 2019*, pp. 1539–1549, (see also [arXiv:1805.04131v2\[cs.DM\]](https://arxiv.org/abs/1805.04131)).