The Path-TSP: Two Solvable Cases

Eranda Çela[1], Vladimir G. Deineko[2], Gerhard J. Woeginger[3]

[1] Department of Discrete Mathematics, TU Graz, Austria
`cela@math.tugraz.at`

[2] Warwick Business School, The University of Warwick, United Kingdom
`Vladimir.Deineko@wbs.ac.ukPythy@MagnaGraeciaUniversity.it`

[3] Computer Science 1 (Algorithms and Complexity), RWTH Aachen University, Germany
`woeginger@cs.rwth-aachen.de`

Abstract

In the Path-TSP, the travelling salesman is looking for the shortest (s, t)-TSP-path, i.e. a path through all cities of a given set of cities starting at a given city s and ending at another given city t, $s \neq t$, after visiting every city exactly once. In this paper we identify two new polynomially solvable cases of the Path-TSP where the distance matrix of the cities is a Demidenko matrix or a Van der Veen matrix, respectively. In each case we characterize the combinatorial structure of optimal (s, t)-TSP-paths and use the obtained results to generate dynamic programming algorithms for these problems. Given the number n of the cities our algorithms have a time complexity of $O(|t - s|n^5)$ in the case of a Demidenko distance matrix and $O(n^3)$ in the case of a Van der Veen distance matrix.

Keywords : *combinatorial optimization, dynamic programming, Path-TSP, tractable special case, Demidenko matrix, Van der Veen matrix.*

1 Introduction

Given n cities with an $n \times n$ symmetric matrix of distances between them, as well as two specific cities s and t, $s \neq t$, the Path-TSP consists in finding a shortest path starting at city s, ending at city t and visiting each city exactly once. Due to the symmetry of the distance matrix it is enough to consider only (s, t)-TSP-paths with $s < t$. Thus any (s, t)-TSP-path can be thought as being directed: it starts at the city with the smaller index s and ends at the city with the larger index t, for any $s, t \in \{1, 2, \dots, n\}$, $s < t$. Analogous to the classical TSP also the Path-TSP is known to be NP-hard and APX-hard, even in the metric case, see eg. [6]. Given the hardness of the problem the characterisation of special cases which can be solved in polynomial time as well as the development of approximation algorithms are of obvious interest. Very recently Zenklusen [6] proposed a new 1.5-approximation algorithm for the metric Path-TSP.

In this paper we focus on polynomially solvable cases (p.s.c.) of the Path-TSP which arise when the distance matrix of the cities has specific algebraic properties. In contrast to an impressive number of p.s.c. of the TSP (see eg. surveys [1, 3] and the references therein), we are aware of just a single p.s.c. of the Path-TSP, namely the case where the cities are points in the Euclidean plane such that each city is located on the border of the convex hull of the set of cities. In 1998 Garcia, Jodrá and Tejel ([2]) gave a linear time algorithm to solve this problem. In general, the combinatorial structure of optimal solutions of a p.s.c. of the Path-TSP for a specific class of distance matrices can be essentially different from the combinatorial structure of optimal solutions of p.s.c. of the TSP for the same special class of distance matrix. Thus, in general, p.s.c. of the Path-TSP cannot be obtained as a modification of already known p.c.s. of the TSP, as also confirmed by the result of [2].

Our contribution. We identify two new p.s.c. of the Path-TSP related to distance matrices with specific algebraic properties which are formulated in terms of inequalities to be fulfilled by the entries of the matrices. In particular we show that the Path-TSP is polynomially solvable if the distance matrix of the given cities is (a) a Demidenko matrix or (b) a Van der Veen matrix. Both matrix classes are defined in Section 2. The basic idea of our approach is to first investigate the combinatorial structure of optimal (s,t)-TSP-paths and identify certain properties P such that there always exists an optimal solution fulfilling these properties. In a second step we show that it is possible to optimize in polynomial time over the set of (s,t)-TSP-paths which fulfill the properties P.

2 Definitions and Notations

Definition 1 *Consider a set of n cities $\{1,2,\ldots,n\}$ with a symmetric $n\times n$ distance matrix $C=(c_{ij})$. Given a natural number k, $k\le n$, an* (s,t)-path *is a sequence $\tau=\langle\tau_1=s,\tau_2,\ldots,\tau_k=t\rangle$ of cities which starts at s, ends at t, and contains every city from $\{\tau_1,\tau_2,\ldots,\tau_k\}\subseteq\{1,2,\ldots,n\}$ exactly once. If $k=n$, the (s,t)-path τ is an (s,t)-TSP-path. For $i\in\{1,2,\ldots,k-1\}$ the successor of τ_i in τ is denoted by $\tau(\tau_i)$, i.e. $\tau_{i+1}=\tau(\tau_i)$. For $i\in\{1,2,\ldots,k-1\}$, an arc (τ_i,τ_{i+1}) in τ is called an increasing arc (decreasing arc) iff $\tau_i<\tau_{i+1}$ ($\tau_i>\tau_{i+1}$).*

An (s,t)-path τ is called λ-pyramidal *(*ν-pyramidal*) if it starts with a subsequence of increasing (decreasing) arcs followed by a subsequence of decreasing (increasing) arcs while the original sequence is just the concatenation of these two subsequences.*

For $i\in\{1,\ldots,k\}$, a city τ_i in a path τ is called a peak *(*valley*) in τ if the arc connecting it with its predecessor in τ is an increasing (decreasing) arc and the arc connecting it with its successor in τ is a decreasing (increasing) arc, where the monotonicity requirements need to be fulfilled only if the corresponding arcs exist.*

Definition 2 *A symmetric $n\times n$ matrix $C=(c_{ij})$ is called a* Demidenko matrix *if the following so-called* Demidenko conditions *are fulfilled*

$$c_{ij}+c_{kl}\le c_{jl}+c_{ki},\qquad \text{for all } 1\le i<j<k<l\le n. \tag{1}$$

A symmetric matrix $C=(c_{ij})$ is called a Van der Veen matrix *if*

$$c_{ij}+c_{kl}\le c_{il}+c_{kj},\qquad \text{for all } 1\le i<j<k<l\le n. \tag{2}$$

3 The Path-TSP on Demidenko distance matrices

An important concept we work with is that of a *forbidden pair of arcs.*

Definition 3 *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.*

The following result is obtained by applying exchange arguments.

Lemma 3.1 *Consider a Path-TSP instance with a Demidenko distance matrix. An optimal (s,t)-TSP-path can be found among the paths that do not contain forbidden pairs of arcs.*

Next we distinguish three cases for the first city s and the last city t in the path: Case (a) $s=1$ and $t=n$, Case (b) $s=1$ and $1<t<n$, and Case (c) $s,t\in\{2,\ldots,n-1\}$, $s<t$.

Case (a). As an consequence of Lemma 3.1 and the observation that any $(1,n)$-TSP-path with a peak other than n contains a pair of forbidden arcs we obtain the following result.

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

Case (b). The following lemma summarizes some structural properties of optimal 1-t-TPS-paths for $t \in \{2, \ldots, n-1\}$, in the case of Demidenko distance matrices.

Lemma 3.2 *Consider a Path-TSP on n cities with a Demidenko distance matrix. For any $t \in \{2, \ldots, n\}$, an optimal $(1,t)$-TSP-path can be found among $(1,t)$-TSP-paths where the peaks decrease and the valleys increase when moving from s to t in the path, i.e. if peak p (valley v) is reached earlier than peak p' (valley v') in the path, than $p > p'$ ($v < v'$) holds.*

Consider now an optimal $(1,t)$-TSP-path τ without forbidden pairs of arcs such that the peaks decrease and the valleys increase when moving from s to t in the path. Let m_1 and m_2, $m_1 > m_2$, be two consecutive peaks. 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:

(i) The (w_1, m_1)-subpath of τ contains no vertex i for which $w_2 < i < m_2$ is fulfilled,

(ii) The (m_1, w_2)-subpath of τ contains no vertex j for which $w_3 < j < m_2$ is fulfilled.

We show that in a path τ fulfilling the properties described in Lemma 3.2 the vertices $m_2+1, m_2+2, \ldots, m_1$ are placed on consecutive positions and form a λ-pyramidal subpath of τ. This leads to a particular structure of such a $(1,t)$-TSP-path; it can be seen as a concatenation of λ-pyramidal subpaths, ν-pyramidal subpaths and a $(1,t)$-TSP path over the cities $\{j, j+1 \ldots, n\}$ for some $j \leq t-2$, where the later path fulfills the properties described in Lemma 3.2. This structure lends itself to a dynamic programming algorithm and we obtain the following result (the details are omitted for the sake of brevity):

Theorem 2 *For a given set of n cities $\{1, 2, \ldots, n\}$ with a Demidenko distance matrix and $t \in \{2, \ldots, n-1\}$, an optimal $(1,t)$-TSP-path can be found in $O(n^5)$ time.*

Case (c). We show (and exploit) the following additional structural properties of an optimal (s,t)-TSP-path.

Lemma 3.3 *There is an optimal (s,t)-TSP-path τ with $1 < s < t < n$, where*

(i) vertex 1 precedes vertex n,

(ii) every vertex in the subpath from s to 1 is smaller than every vertex in the subpath from n to t,

(iii) there is a city $p \in \{s+1, \ldots, t\}$ such that τ can be represented as a concatenation of two paths $\tau_1(p)$ and $\tau_2(p)$ with $\tau_1(p)$ starting at s and visiting all vertices from the set $\{1, 2, \ldots, p-1\}$, and path $\tau_2(p)$ ending at t and visiting all vertices from the set $\{p, \ldots, n\}$.

Now the length of an optimal (s,t)-TSP-path with the structure described by Lemma 3.3 can be computed by considering all possible realisations of vertex $p \in \{s+1, \ldots, t\}$ and the end vertex of $\tau_1(p)$ (or equivalently the start vertex of $\tau_2(p)$) in Lemma 3.3 and using the result of Case (b) to compute the optimal length of the paths $\tau_1(p)$ and $\tau_2(p)$, respectively. Summarizing we obtain the following general result (the detail are omitted for the sake of brevity):

Theorem 3 *For a given a set of n cities $\{1, 2, \ldots, n\}$ with a Demidenko distance matrix and $s, t \in \{2, \ldots, n-1\}$, $s < t$, an optimal (s,t)-TSP-path can be found in $O(|t-s|n^5)$ time.*

4 The Path-TSP on Van der Veen distance matrices

Also in this case we need an appropriate definition of a *forbidden pair of arcs*.

Definition 4 *In an (s,t)-path τ a pair of arcs (i,k) and (j,l), with $k = \tau(i)$ and $l = \tau(j)$ is called a* forbidden *pair of arcs if one of the two conditions (A) or (B) below holds:*

(A) $i < j < l < k$ *or* $i > j > l > k$

(B) $i < j < j+1 < k < l$ *or* $l < k < k+1 < j < i$

The following lemma summarizes some structural properties of optimal (s, t)-TSP-path in the case of Van der Veen distance matrices.

Lemma 4.1 *Consider a Path-TSP instance over the cities* $\{1, 2, \ldots, n\}$ *with a Van der Veen distance matrix* $C = (c_{ij})$ *and* $s, t \in \{1, 2, \ldots, n\}$, $s < t$. *An optimal* (s, t)*-TSP-path can be found among the* (s, t)*-TSP-paths which*

(i) *do not contain forbidden pairs of arcs,*

(ii) *do not contain three arcs* (i_1, j_1), (i_2, j_2), *and* (i_3, j_3) *such that* $\max\{i_1, i_2, i_3\} < \min\{j_1, j_2, j_3\}$,

(iii) *do not contain four arcs* (i_1, j_1), (i_2, j_2), (k_1, l_1) *and* (k_2, l_2), *with* $i_1 < j_1$, $i_2 < j_2$, $k_1 > l_1$, $k_2 > l_2$, *such that* $\max\{i_1, i_2, l_1, l_2\} < \min\{j_1, j_2, k_1, k_2\}$.

The results of Lemma 4.1 lead to a straightforward $O(n^4)$ dynamic programming algorithm to compute an optimal (s, t)-TSP-path in this case (using similar ideas as in the dynamic programming computation of pyramidal tours, see e.g. [4].)

By identifying and exploiting further combinatorial properties of optimal $(1, n)$-TSP-paths we obtain an $O(n^3)$ dynamic programming algorithm. More precisely, we introduce three classes of specially structured subpaths: *long zigzag*, *short zigzag*, *splitted zigzag* and *zigzag* paths. Then we consider the cases (a), (b) and (c) analogously as in the case of Demidenko distance matrices. In Case (a) we show that an optimal $(1, n)$-TSP-path fulfilling the conditions of Lemma 4.1 can be found among specific concatenations of short zigzags and long zigzags, all of which can be represented as edges of an auxiliary weighted digraph with $O(n)$ vertices. Then an optimal $(1, n)$-TSP-path, is computed by solving a single source shortest paths problem in the auxiliary digraph, leading to an $O(n^2)$ time algorithm. In Case (b) we show that an optimal $(1, t)$-TSP-path can be represented as a concatenation of a zigzag path over the cities $\{1, 2, \ldots, t-1\}$ with a λ-pyramidal subpath over the remaining cities. An optimal patching of those subpaths can be done in $O(n^2)$ time by means of dynamic programming. Finally, in Case (c) also the splitted zigzag paths are needed; an optimal (s, t)-TSP-path is a concatenation of splitted zigzags, λ-pyramidal subpaths and/or ν-pyramidal subpaths, as well as of optimal (q, t)-TSP-path over the cities $\{q, q+1, \ldots, t\}$ for some $q < t$. in this case an optimal patching of all subpaths can be determined in $O(n^3)$ time by means of dynamic programming.

5 Conclusions and perspectives

We have considered the Path-TSP problem for two classes of distance matrices, Demidenko matrices and Van der Veen matrices. For each of the two classes we have defined so-called forbidden pairs of arcs and have shown that there always exists an optimal solution which does not contain forbidden pairs of arcs. Further, we have identified a number of combinatorial properties of TSP-paths without forbidden pairs of arcs for each of the two classes. These properties lead to a full characterization of TSP-paths without forbidden pairs of arcs and allow the computation of optimal TSP-paths without forbidden pairs of arcs by means of dynamic programming. In the case of a Demidenko distance matrix an optimal (s, t)-TSP-path over cities $\{1, 2, \ldots, n\}$ can be determined in $O(|t-s|n^5)$ time, where $s, t \in \{1, 2, \ldots, n\}$, $s \neq t$. In the case of Van der Veen matrices we obtain an $O(n^2)$ algorithm for $s = 1$ and $t \in \{2, \ldots, n\}$, and an $O(n^3)$ algorithm for $s, t \in \{2, \ldots, n\}$, $s \neq t$.

The new polynomially solvable special cases can be used to define exponential neighbourhoods for the Path-TSP over which it can be optimized in polynomial time, see e.g. [5] for a discussion of similar approaches in the case of the TSP. The design of local search algorithms for the Path-TSP based on these exponential neighborhoods and the analysis of their performance remains an open problem for further investigations.

References

[1] Vladimir G. Deineko, Bettina Klinz, Alexander Tiskin, and Gerhard J. Woeginger. Four-point conditions for the TSP: the complete classification. *Discrete Optimization*, 14:147–159, 2014.

[2] Alfredo Garcia, Pedro Jodrá, and Javier Tejel. An efficient algorithm for on-line searching in Monge path-decomposable tridimensional arrays. *Information Processing Letters*, 68:3–9, 1998.

[3] Santosh N. Kabadi. Polynomially solvable cases of the TSP. In Gregory Gutin and Abraham P. Punnen, editors, *The travelling salesman problem and its variations*, chapter 11, pages 489–583. Kluwer Academic Publishers, 2002.

[4] Yoshiaki Oda and Katsuhiro Ota. Algorithmic aspects of pyramidal tours with restricted jump-backs. *Intersciplinary Information Sciences*, 7:123–133, 2001.

[5] James B. Orlin and Dushyant Sharma. The extended neighborhood: Definition and characterization. *Mathematical Programming*, 101:537–559, 2004.

[6] Riko Zenklusen. A 1.5-approximation for path TSP. In *SODA '19: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1539–1549. 2019.