Operations Research Letters 55 (2024) 107130

Contents lists available at ScienceDirect

Operations Research Letters

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

The bipartite travelling salesman problem: A pyramidally solvable case

Vladimir G. Deineko [a], Bettina Klinz [b,∗], Gerhard J. Woeginger [c,1]

[a] Warwick Business School, The University of Warwick, Coventry CV4 7AL, United Kingdom
[b] Institute of Discrete Mathematics, TU Graz, Austria
[c] Department of Computer Science, RWTH Aachen, Germany

ARTICLE INFO

Article history:
Available online 27 May 2024

Keywords:
Combinatorial optimisation
Bipartite travelling salesman problem
Pyramidally solvable case
63605ec6eb4e243dcb2eeb7d Van der Veen matrix
Recognition algorithm

ABSTRACT

Given k blue cities, k red cities and a $2k \times 2k$ distance matrix, the task in the bipartite travelling salesman problem is to find a shortest tour which alternately visits blue and red cities. We consider the special case of Van der Veen distance matrices and show that it remains NP-hard in general but can be solved in $O(k^2)$ time when all vertices with odd indices are blue and all with even indices are red.

1. Introduction

In the *travelling salesman problem* (TSP) a salesperson is looking for the shortest tour to visit all cities from a given list of cities. The input consists of n cities (including the city where the salesperson lives) and an $n \times n$ distance matrix $C = (c_{ij})$ where c_{ij} provides the distance (or travel time) from city i to city j. Throughout this paper we assume that distance matrices are *symmetric*, but we do neither require non-negativity nor the triangle inequality. The task in the TSP is to find the shortest tour visiting all the cities.

The TSP is probably one of the best studied NP-hard optimisation problems and has served as important benchmark problem in discrete optimisation with a long list of outstanding contributions to the theory and practice of the field (see e.g. the monographs [1,23,30]). One of the well established directions of research for NP-hard optimisation problems is the investigation of polynomially solvable special cases (see the surveys [10,18,22,26] for further references).

In the *bipartite travelling salesman problem* (BTSP) the set of cities $\{1, \ldots, n\}$ is partitioned into two subsets, the set K_1 of blue cities and the set K_2 of red cities where $|K_1| = |K_2| = k$, $n = 2k$. Any feasible tour in the BTSP has to alternate between blue and red cities. The objective is to find the shortest such tour. Note that by defining the distances between cities of the same colour to be infinite, the BTSP can be considered as special case of the TSP.

Motivation and previous work. An important application of the BTSP can be found in the context of container terminal management (see [7–9,13,29]). In a container terminal trains with containers arrive to a terminal and have to be unloaded to a storage area. The containers have fixed positions on the trains and the unloading is performed by a single crane. The goal is to minimise the unloading time. The special case with only k storage positions specified for the locations of k containers from the train, can be modelled as BTSP. The BTSP has also drawn the attention of researchers ([4,5,14,19]) due to its relevance to pick-and-place (or grasp-and-delivery) robots ([2,3,28,31,25]).

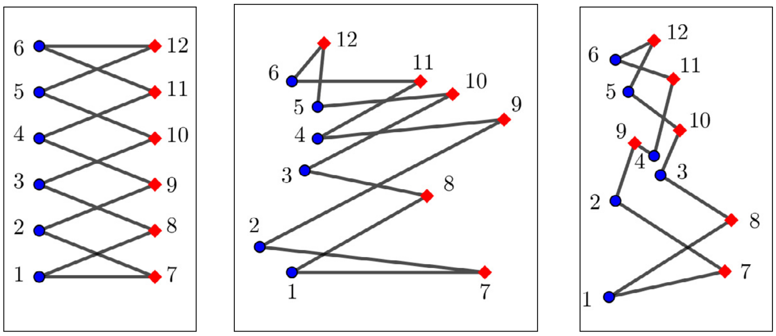
The BTSP is not as well studied as the TSP. In particular, while there are plenty of publications on polynomially solvable cases of the TSP, we are aware of only a few papers [17,20,24,32] published on solvable cases of the BTSP. Note that the above mentioned transformation from the BTSP to the TSP typically will not preserve special structures present in the BTSP distance matrix implying that efficiently solvable cases of the BTSP are not a simple by-product of such cases for the TSP. Moreover, the TSP polytope differs considerably from the BTSP polytope, cf. [27].

Halton [24] was the first who provided a polynomially solvable case of the BTSP. He considered the shoelace problem where cities represent the eyelets of shoes and the objective is to find an optimal shoe lacing strategy that minimises the length of the shoelace. In Halton's model the eyelets can be viewed as points in the Euclidean plane: the blue points $K_1 = \{1, 2, \ldots, k\}$ have coordinates $(0, d), (0, 2d), \ldots, (0, kd)$ and the red points $K_2 = \{k + 1, k + 2, \ldots, n\}$ have coordinates $(a, d), (a, 2d), \ldots, (a, kd)$, respectively. Halton proved that the optimal BTSP solution in this case has a special structure which is illustrated in Fig. 1(a).

∗ Corresponding author.
E-mail addresses: Vladimir.Deineko@wbs.ac.uk (V.G. Deineko), klinz@math.tugraz.at (B. Klinz).

[1] Deceased April 1, 2022.

https://doi.org/10.1016/j.orl.2024.107130



(a) (b) (c)

Fig. 1. Illustration from [17] for polynomially solvable BTSP cases as variants of the shoelace problem: (a) - Halton [24] case; (b) - Misiurewicz [32] case; (c) case of Deineko & Woeginger [17]. Blue vertices are displayed as blue circles and red vertices as red squares.

The shoelace problem is a nice interpretation of the BTSP that can be used for entertaining and educational purposes. Misiurewicz [32] argued that Halton's case is based on quite restricted assumptions which are hardly met in real life. He generalized Halton's model to the case referred to as "for old shoes" (Fig. 1(b)) which arises for distance matrices C which fulfill

$$c_{ij} + c_{\ell m} \le c_{im} + c_{\ell j}$$
$$\text{for all } (i, \ell, j, m) \quad \text{s.t.} \quad 1 \le i \le \ell \le k < j \le m \le n.$$

Deineko and Woeginger [17] went on further and investigated the case "for *very old* shoes" (Fig. 1(c)).

Notice that the cities in Halton's case are points in the Euclidean plane and are placed on the boundary of their convex hull. The blue and the red points occur consecutively along this boundary. In [20] an even more general case of the Euclidean BTSP is considered where the points are still on the boundary of their convex hull, but the points with the same colour are not necessarily on consecutive positions. For this case, there exists a candidate set of exponential size within which the optimal BTSP tour can be found. The running time of the resulting algorithm is $O(n^6)$.

Recently the bipartite TSP-path problem with the points placed on a line got studied in [6] where several cases are described which can be solved in linear time.

The special BTSP cases considered in [6,20,24] have been characterised in terms of special locations of points in the Euclidean plane, while in [17] and [32] the considered special cases are obtained by imposing conditions on the entries in the distance matrices. These conditions are defined by sets of linear inequalities. This approach is widely used in the research literature on polynomially solvable cases of the TSP (see e.g. [18] and the references therein).

Contribution and organisation of the paper. In this paper we investigate the BTSP on Van der Veen distance matrices (see conditions (1) in Section 2 for a definition) and the newly introduced class of relaxed Van der Veen matrices which result if certain of the linear inequalities which are involved in the definition of Van der Veen matrices are dropped (for a definition see conditions (4) in Section 3).

The class of Van der Veen matrices has been well investigated in the literature on polynomially solvable cases of the TSP, but has not been considered in the context of the BTSP. We first show that the BTSP when restricted to Van der Veen distance matrices remains NP-hard. Then we show that the even-odd BTSP which results if K_1 contains all cities with odd indices and K_2 contains all cities with even indices becomes solvable in polynomial time when restricted to (relaxed) Van der Veen distance matrices. In this case an optimal tour can be found within the set of pyramidal tours in $O(n^2)$ time.

We can go one step further. We can recognise the class of matrices C which become relaxed Van der Veen matrices after renumbering the cities in K_1 and in K_2 with independent permutations which allows us to find an optimal tour for this subclass of permuted relaxed Van der Veen matrices in $O(n^4)$ time (the time needed by the recognition algorithm).

In Section 2 we provide the definitions and preliminaries needed in the rest of the paper. Section 3 constitutes the heart of the paper and contains both the hardness result for the BTSP restricted to general Van der Veen matrices as well as our polynomial time algorithm for the even-odd BTSP restricted to relaxed Van der Veen matrices. Section 4 describes a polynomial time recognition algorithm for a subclass of permuted relaxed Van der Veen matrices.

2. Definitions and preliminaries

Given an $n \times n$ distance matrix $C = (c_{ij})$ the objective in the TSP is to find a cyclic permutation τ of the set $\{1, 2, \ldots, n\}$ that minimises the travelled distance $c(\tau) = \sum_{i=1}^{n} c_{i\tau(i)}$.

In the context of the TSP, the cyclic permutations are also called *tours*, the elements of $\{1, 2, \ldots, n\}$ are called *cities* or *points*, and $c(\tau)$ is referred as the length of the tour τ. Let $\mathcal{C}_n$ denote the set of all cyclic permutations over $\{1, 2, \ldots, n\}$. For $\tau \in \mathcal{C}_n$, we denote by τ^{-1} the *inversion* of τ, i.e., the permutation for which $\tau^{-1}(i)$ is the predecessor of i in τ, for $i = 1, \ldots, n$. We also use the cyclic representation of a tour τ in the form

$$\tau = \langle i, \tau(i), \tau(\tau(i)), \ldots, \tau^{-1}(\tau^{-1}(i)), \tau^{-1}(i), i \rangle.$$

In the bipartite TSP (BTSP) on top of an $n \times n$ distance matrix C, we are also given a partition of the $n = 2k$ cities into the two sets K_1 and K_2 with $K_1 \cup K_2 = \{1, 2, \ldots, n\}$ and $|K_1| = |K_2| = k$. The special case of the *even-odd BTSP* results when K_1 contains all cities with odd indices and K_2 contains all cities with even indices.

The set $\mathcal{T}_n(K_1, K_2)$ of all feasible tours for the BTSP can formally be defined as

$$\mathcal{T}_n(K_1, K_2) = \{\tau \in \mathcal{C}_n \mid \tau^{-1}(i), \tau(i) \in K_2$$
$$\text{for } i \in K_1;\ \tau^{-1}(i), \tau(i) \in K_1 \text{ for } i \in K_2\}.$$

We will refer to the tours in $\mathcal{T}_n(K_1, K_2)$ as *bipartite* tours or feasible BTSP tours. For example, if $K_1 := \{1, 2, \ldots, k\}$ and $K_2 := \{k+1, \ldots, n\}$, then the tour

$$\tau^* = \langle 1, k+1, 2, k+3, 4, k+5, 6 \ldots, 7, k+6, 5, k+4, 3, k+2, 1 \rangle$$

which is illustrated in Fig. 1 (a) is a feasible BTSP tour, i.e., a member of $\mathcal{T}_n(K_1, K_2)$.

Let $C[K_1, K_2]$ denote the $k \times k$ matrix which is obtained from matrix C by *deleting* all rows with indices from K_2 and all columns with indices from K_1. Clearly, the length $c(\tau)$ of any feasible BTSP tour τ is calculated by using *only* entries from $C[K_1, K_2]$.

A tour $\tau = \langle 1, \tau_2, \ldots, \tau_m, n, \tau_{m+2}, \ldots, \tau_{n-2}, 1\rangle$ is called a *pyramidal tour*, if $1 < \tau_2 < \ldots < \tau_m < n$ and $n > \tau_{m+2} > \ldots > \tau_{n-2} > 1$. An instance of the TSP/BTSP is called *pyramidally solvable* if an optimal solution to the instance can be found within the set of pyramidal tours.

The notion of pyramidal tours is well known in the rich literature on polynomially solvable cases of the TSP (see the surveys [10,22,23,26,18]). Although the set of pyramidal tours contains $\Theta(2^n)$ tours, a shortest pyramidal tour can be found in $O(n^2)$ time by dynamic programming (see e.g. Section 7 in [22]).

A symmetric $n \times n$ matrix C is called a *Van der Veen* matrix if it fulfils the so-called *Van der Veen conditions*

$$c_{ij} + c_{j+1,m} \le c_{im} + c_{j+1,j} \qquad \text{for all } 1 \le i < j < j+1 < m \le n. \tag{1}$$

While conditions like Halton's condition can be easily visualized, it is difficult to describe point sets which lead to Van der Veen distance matrices [15]. Note that the none of the distance matrices that results from the points sets displayed in Fig. 1 is a Van der Veen matrix. (The inequality (1) is violated e.g. for $i = 1$, $j = 3$ and $m = 7$.)

3. Complexity results for the BTSP on (relaxed) Van der Veen matrices

In this section we investigate the complexity of the BTSP restricted to Van der Veen matrices and relaxed Van der Veen matrices. It was proved by van der Veen [34] back in 1994 that the TSP with a distance matrix that satisfies conditions (1) is pyramidally solvable. For the BTSP the situation is more complex. It turns out that the partitioning into blue and red cities which is part of the input plays a crucial role.

We will show that the BTSP restricted to Van der Veen matrices remains NP-hard while the even-odd BTSP where the colouring of the cities is chosen according to the parity of the city indices becomes polynomially solvable for Van der Veen matrices and even for relaxed Van der Veen matrices.

We start with the hardness result for the special case where the first k vertices are coloured blue and the remaining ones are coloured red.

Theorem 3.1. *The BTSP is NP-hard on $n \times n$ Van der Veen distance matrices when the $n = 2k$ cities are partitioned into the sets $K_1 := \{1, 2, \ldots, k\}$ and $K_2 := \{k+1, \ldots, 2k\}$.*

Proof. The proof follows along the lines of the proofs of similar results for the TSP (see [16] and [33]) and makes some adjustments required for the BTSP. The proof is done by a reduction from the NP-hard HAMILTONIAN CYCLE PROBLEM IN BIPARTITE GRAPHS (cf. [21]).

Let $G = (K_1 \cup K_2, E)$ be a bipartite graph with $E \subset K_1 \times K_2$. From G we construct a $2k \times 2k$ Van der Veen matrix $C = (c_{ij})$ as follows. The items in $C[K_1, K_2]$ are obtained from the adjacency matrix of G: If there is an edge between $i \in K_1$ and $j \in K_2$, we set $c_{ij} = c_{ji} = 0$, otherwise we set $c_{ij} = c_{ji} = 1$.

Notice that any tour for the BTSP involves only trips between cities which are not on the same side of the partition. The corresponding distances are elements of the submatrix $C[K_1, K_2]$. Hence the graph G contains a Hamiltonian cycle if and only if the length of an optimal solution to the BTSP instance with matrix C is 0.

What is left to be shown is that the yet undefined elements in C can be set in a way such that the resulting matrix is a Van der Veen matrix. This can be achieved as follows. For $i, j \in K_1 = \{1, 2, \ldots, k\}$, $i < j$, we set $c_{ij} = -(k+1) + j$, $c_{ji} = c_{ij}$ and we set $c_{k+i,k+j} = -i$, $c_{k+j,k+i} = c_{k+i,k+j}$. This completes the construction of the matrix C.

We now need to check that the inequalities (1) are fulfilled to confirm that C is indeed a Van der Veen matrix. The following five cases need to be considered.

(a) $k \in [m, n]$: In this case all matrix entries in (1) belong to $C[K_1, K_1]$ as $k \le m$. Plugging in the values as set above yields $-(k+1) + j - (k+1) + m \le -(k+1) + m - (k+1) + j + 1$ which always holds true.

(b) $k \in [j+1, m-1]$: In this case c_{ij} and $c_{j+1,j}$ belong to $C[K_1, K_1]$, and $c_{j+1,m}$ and c_{im} to $C[K_1, K_2]$. Then (1) can be rewritten as $-(k+1) + j + c_{j+1,m} \le c_{im} - (k+1) + j + 1$, or $c_{j+1,m} \le c_{im} + 1$, which is true since all entries in $C[K_1, K_2]$ are either 0 or 1 by construction.

(c) $k \in [j, j]$: In this case c_{ij} (i.e. c_{ik}) belongs to $C[K_1, K_1]$, hence $c_{ij} = -(k+1) + k = -1$. Moreover, $c_{k+1,k}$ and c_{im} are 0-1 valued and $c_{k+1,m} = -1$ since it belongs to $C[K_2, K_2]$. The inequality $-1 - 1 \le c_{k+1,k} + c_{im}$ is always satisfied.

(d) $k \in [i, j-1]$: In this case c_{ij} and c_{im} belong to $C[K_1, K_2]$ (hence are 0-1 valued) and $c_{j+1,m}$ and $c_{j+1,j}$ belong to $C[K_2, K_2]$. The resulting inequality to be checked is $c_{ij} - (j + 1 - k) \le c_{im} - (j - k)$ which is trivially satisfied.

(e) $k \in [1, i-1]$: All entries involved in (1) are from $C[K_2, K_2]$. The inequality to be checked is $-(i-k) - (j+1-k) \le -(i-k) - (j-k)$ which is trivially fulfilled as $-1 \le 0$. $\square$

The situation becomes more favourable for the even-odd BTSP. Remember that there we consider $K_1 := \{1, 3, \ldots, 2k-1\}$ and $K_2 := \{2, 4, \ldots, 2k\}$ with $n = 2k$. The following definition and lemma will turn out to be helpful in the proof of our result for the even-odd BTSP.

Let $\Delta(i, l, j, m) := c_{ij} + c_{lm} - c_{im} - c_{jl}$. Note that for a symmetric matrix C, we have $\Delta(i, l, j, m) = \Delta(j, m, i, l)$.

Lemma 3.2. *Let C be a symmetric $n \times n$ matrix and $\Delta(i, j, l, m)$ be as defined above.*

(a) The Van der Veen inequalities (1) are equivalent to

$$\Delta(i, j+1, j, m) \le 0 \qquad \text{for } 1 \le i < j < j+1 < m \le n \tag{2}$$

which by symmetry is equivalent to

$$\Delta(j, m, i, j+1) \le 0 \qquad \text{for } 1 \le i < j < j+1 < m \le n. \tag{3}$$

(b) $\Delta(i_1, i_2, j_1, j_2) = \Delta(i_1, i', j_1, j_2) + \Delta(i', i_2, j_1, j_2)$ holds for $1 \le i_1 < i' < i_2 \le n$, $1 \le j_1 < j_2 \le n$.

(c) $\Delta(i_1, i_2, j_1, j_2) = \Delta(i_1, i_2, j_1, j') + \Delta(i_1, i_2, j', j_2)$ holds for $1 \le i_1 < i_2 \le n$, $1 \le j_1 < j' < j_2 \le n$.

Proof. (a): Direct consequence of the definition of Δ.

(b): Note that $\Delta(i_1, i', j_1, j_2) + \Delta(i', i_2, j_1, j_2) = c_{i_1 j_1} + c_{i' j_2} - c_{i_1 j_2} - c_{i' j_1} + c_{i' j_1} + c_{i_2 j_2} - c_{i' j_2} - c_{i_2 j_1} = c_{i_1 j_1} + c_{i_2 j_2} - c_{i_1 j_2} - c_{i_2 j_1} = \Delta(i_1, i_2, j_1, j_2)$.

(c): Follows from (b) by symmetry $(\Delta(i, l, j, m) = \Delta(j, m, i, l))$. $\square$

The lemma below will be essential to prove the correctness of the tour improvement technique used subsequently to prove Theorem 3.4.

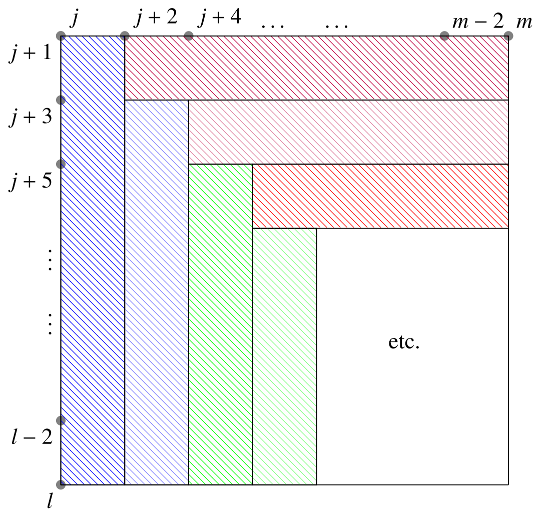


Fig. 2. Schematic representation of the decomposition (6).

Lemma 3.3. *Let C be an $n\times n$ Van der Veen matrix. Then*

$$c_{j+1,j}+c_{lm}-c_{jl}-c_{j+1,m}\le 0 \qquad \text{for all } (j,l,m)\in J \tag{4}$$

with $J=\{(j,l,m)\in\mathbb{N}^3\mid 1\le j<j+1<l,m\le n;\ j\equiv m \mod 2,\ j+1\equiv l \mod 2\}$.

Proof. Note that the inequalities (4) can be rewritten as $\Delta(j+1,l,j,m)\le 0$ for all $(j,l,m)\in J$. Let $q=j+1$. It hence suffices to prove the following two statements.

- Statement I: $\Delta(q,l,j,m)\le 0$ for all $(j,l,m)\in J$ with $\boldsymbol{l<m}$
- Statement II: $\Delta(j+1,l,j,m)\le 0$ for all $(j,l,m)\in J$ with $\boldsymbol{m<l}$

To allow us to use induction, we will prove Statement I for $q=j+1,j+3,\dots,l-2$.

Proof of Statement I. As $j+1<l<m$ and $l\equiv j+1 \mod 2$, we have $l\ge j+3$. Let $l=j+2L+3$ with $L\ge 0$.

- Induction basis: $q=l-2=j+2L+1$. Lemma 3.2 (use (c)) yields

  $$\begin{aligned}\Delta(q,l,j,m)&=\Delta(j+2L+1,j+2L+3,j,m)\\&=\Delta(j+2L+1,j+2L+3,j,j+2L+2)\\&\quad+\Delta(j+2L+1,j+2L+3,j+2L+2,m).\end{aligned}$$

  Lemma 3.2 (use (a)) implies that both terms in the sum are ≤ 0. Hence the base case is established.
- Induction step: Assume Statement I is true for $q=p$. We want to show it for $q=p-2$.

  $$\begin{aligned}\Delta(p-2,l,j,m)&=\Delta(p-2,p,j,m)+\Delta(p,l,j,m)\\&=\Delta(p-2,p,j,p-1)\\&\quad+\Delta(p-2,p,p-1,m)+\Delta(p,l,j,m)\end{aligned} \tag{5}$$

  Note that the first two terms in (5) are ≤ 0 due to (2) and (3). The third one is ≤ 0 by the induction hypothesis. Hence $\Delta(p-2,l,j,m)\le 0$.

Proof of Statement II. As $m>j$ and $m\equiv j \mod 2$, we have $m\ge j+2$. Thus $l\ge j+3$. If $m=j+2$, Statement II is true because $\Delta(j+1,l,j,j+2)\le 0$ holds due to (3).

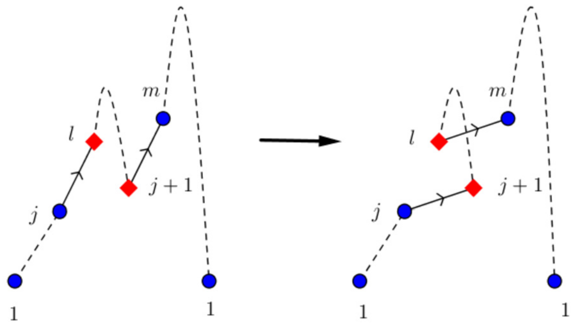


Fig. 3. Illustration of one iteration of the tour improvement technique.

Otherwise let $m=j+2K+2$ with $K\ge 1$. Then $K=(m-j-2)/2$. By a repetitive application of properties (b) and (c) of Lemma 3.2 one obtains the following decomposition (for an illustration see Fig. 2).

$$\begin{aligned}\Delta(j+1,l,j,m)=\sum_{k=1}^{K}\Big\{&\Delta(j+2k-1,l,j+2k-2,j+2k)\\&+\Delta(j+2k-1,j+2k+1,j+2k,m)\Big\}\\&+\Delta(j+2K+1,l,j+2K,m)\end{aligned} \tag{6}$$

Statement II immediately follows from the decomposition above by observing that all terms in the sum are ≤ 0 due to the properties (2) and (3) of a Van der Veen matrix. (Note that the last term equals $\Delta(j+2K+1,l,j+2K,j+2K+2)$.) □

Theorem 3.4. *The even-odd BTSP on $n\times n$ Van der Veen distance matrices is pyramidally solvable and hence can be solved in $O(n^2)$ time.*

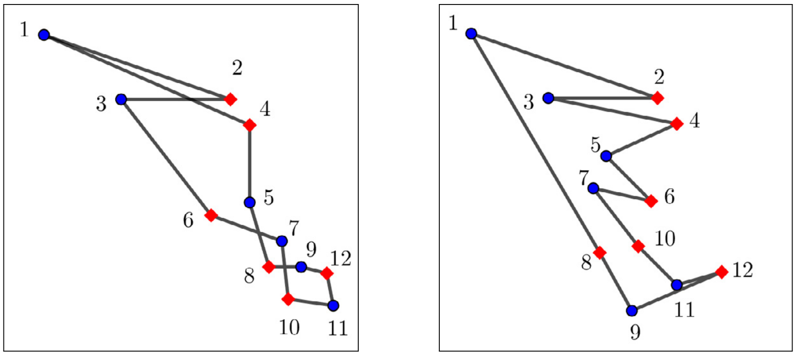
Proof. In this proof the well-known tour improvement technique (cf. [10]) is used. Assume that we are given a bipartite feasible tour on $n=2k$ cities, with the k blue cities placed on odd positions, and the k red cities placed on even positions. We will show how to transform a feasible tour τ into a pyramidal tour which is also a *feasible* BTSP solution, i.e., with the blue cities placed on odd positions, and the red cities placed on even positions.

Index i in tour τ is called a *valley*, if $\tau^{-1}(i)>i$ and $i<\tau(i)$. Observe that a tour is pyramidal if and only if city 1 is its only valley. If tour τ is not a pyramidal tour, we identify the minimal valley which is greater than 1. Let this valley be $j+1$. Then vertex j cannot be a peak and so we can w.l.o.g. assume that $\tau(j)=l>j$ since the distance matrix C is symmetric and we can choose to either work with τ or its inversion τ^{-1}. Furthermore, let $\tau(j+1)=m$. Notice that the cities j and $j+1$ have different parity. Assume that $j+1$ is even (a red city), then j is odd (a blue city).

We now create a new feasible BTSP tour τ' as follows. We delete the edges (j,l), $(j+1,m)$, invert the sub-tour $\langle l,\dots,j+1\rangle$ to obtain the sub-tour $\langle j+1,\dots,l\rangle$, and then introduce two new edges $(j,j+1)$, (l,m). For an illustration see Fig. 3. It is obvious that we obtain this way a new tour which is a feasible solution for the BTSP. Moreover, the minimal valley in the new tour is bigger than $j+1$. After a finite number of iterations we end up with a pyramidal tour which is a feasible solution to the BTSP.

Observe that $c(\tau')=c(\tau)-c_{jl}-c_{j+1,m}+c_{j+1,j}+c_{lm}$. Hence Lemma 3.3 implies that $c(\tau')\le c(\tau)$. Iterative application implies that the procedure described above transforms any feasible BTSP tour into a pyramidal BTSP tour with the same or a smaller total length which concludes the proof. □

Note that in the proof above only the subset (4) of the Van der Veen inequalities is needed. All matrix entries involved in (4) are



Instance A Instance B

Point number	1	2	3	4	5	6	7	8	9	10	11	12
x-coord. in A	5	34	17	37	37	31	42	40	45	43	50	49
y-coord. in A	45	35	35	31	19	17	13	9	9	4	3	8
x-coord. in B	5	35	17	37	26	33	24	25	30	31	37	44
y-coord. in B	45	35	35	31	26	19	21	11	2	12	6	8

Fig. 4. Optimal tours for instances of the even-odd BTSP with a Van der Veen matrix (Instance A) and with a relaxed Van der Veen matrix (Instance B).

entries of the submatrix $C[K_1, K_2]$. We call a matrix C that satisfies the conditions (4) a *relaxed* Van der Veen matrix. The following corollary follows immediately.

Corollary 3.5. *The even-odd BTSP on an $n \times n$ relaxed Van der Veen matrix is pyramidally solvable.*

For the sake of illustration, Fig. 4 shows two set of points in the Euclidean plane along with their corresponding optimal BTSP tours. The distance matrix in instance A is a Van der Veen matrix while the distance matrix in instance B is a relaxed Van der Veen matrix. In instance B we chose the first four points (as in instance A) and randomly generated the other points to agree with conditions (4). In instance B, 17 out of 165 Van der Veen inequalities (1) are violated; e.g., take the inequality that results for $i = 6$, $j = 9$ and $m = 11$.

4. Recognition of a subclass of permuted relaxed Van der Veen matrices

It is obvious that a matrix property which is defined by linear inequalities as it is the case for Van der Veen matrices depends on the numbering of the rows and of the columns. Note that no efficient algorithm for recognising permuted Van der Veen matrices is known. The subclass of Euclidean permuted Van der Veen matrices can be recognized in $O(n^4)$ time, see [12].

For the BTSP the partitioning of the set of cities into the coloured sets also has to be taken into consideration. We henceforth assume that the partitioning of the set of cities into the subsets K_1 and K_2 is given, but we have the freedom of choosing the numbering of the cities in each subset. More specifically, we consider symmetric matrices that satisfy the system of linear inequalities (4) with the partitioning $K_1 := \{1, 3, \ldots, 2k-1\}$ and $K_2 := \{2, 4, \ldots, 2k\}$. We assume that the initial numbering of the cities in K_1 and K_2 was chosen arbitrarily, and the system (4) is not satisfied. We pose the question whether it is possible to efficiently (re)construct a numbering of the cities in K_1 and K_2 such that the properties (4) are fulfilled (at least one such ordering exists by assumption).

To simplify the further notations, we define a new $k \times k$ asymmetric matrix $A := C[K_1, K_2]$ with $a_{ij} = c_{2i-1,2j}$. Using this notation, the system (4) can be rewritten as

$$a_{11} + a_{lm} \le a_{1m} + a_{l1} \quad l, m = 2, 3, \ldots, k, \tag{7}$$

$$a_{i,i-1} + a_{lm} \le a_{im} + a_{l,i-1}$$
$$l = i+1, \ldots, k;\, m = i, \ldots, k;\, i = 2, \ldots, k-1 \tag{8}$$

$$a_{ii} + a_{lm} \le a_{im} + a_{li}$$
$$l = i+1, \ldots, k;\, m = i+1, \ldots, k;\, i = 2, \ldots, k-1 \tag{9}$$

Note that (8) corresponds to the case j even while (7) and (9) together cover the case j odd (these two conditions could be joined into one, but it is more convenient being able to refer to (7) separately below).

Proposition 4.1. *Given a $k \times k$ matrix $A = (a_{ij})$, it can be decided in $O(k^4)$ time whether there exist permutations γ and δ such that the permuted matrix $(a_{\gamma(i)\delta(j)})$ satisfies conditions (7)-(9). If the permutations γ and δ exist, they can be found in time $O(k^4)$.*

Proof. The system (7)-(9) is similar to the systems investigated in [11] and [17]. The proof below follows the logic of the approach used by [17].

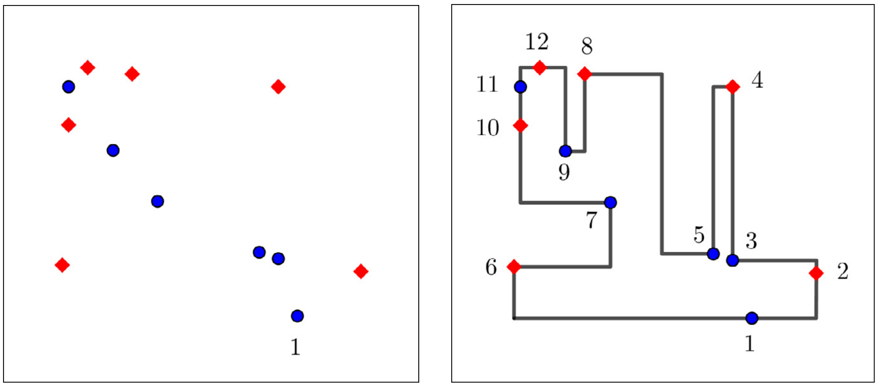
First try all k indices as candidates for the first position in γ. Let $\gamma(1) = 1$. According to (7), index i can be placed at the first position in δ if and only if

$$a_{1i} + a_{st} \le a_{si} + a_{1t} \text{ for all } s \ne 1, t \ne i. \tag{10}$$

If there is another candidate j with the same property ($i \ne j$), then $a_{1i} + a_{sj} = a_{si} + a_{1j}$ needs to hold, i.e., $a_{sj} = a_{si} + d$ for all s, where $d = a_{1j} - a_{1i}$ is a constant for fixed i and j. Since adding a constant to a row or a column of matrix A does not affect the inequalities (7)-(9), any of the indices i or j can be placed at the first position in δ.

We claim that candidate i can be chosen in $O(k^2)$ time. This can be seen as follows. Note that the transformation $a'_{st} = a_{st} - a_{1t}$, $s = 1, \ldots, k$, $t = 1, \ldots, k$, transforms matrix A into matrix A' with zeros in the first row. The inequalities (10) are equivalent to $a'_{st} \le a'_{si}$ for all s, t and i. Hence index i can be found in $O(k^2)$ time by looking through the indices of the maximal entries in the rows of A'.

An index for the second position in δ needs to be chosen by using the same approach based on inequalities (8). An index for the second position in γ needs to be chosen by using the inequalities (9). This approach is going to be repeated for all positions.



(a)

(b)

Point number	1	2	3	4	5	6	7	8	9	10	11	12
x-coord.	38	48	35	35	32	1	16	12	9	2	2	5
y-coord.	8	15	17	44	18	16	26	46	34	38	44	47

Fig. 5. (a) - Set of points which satisfies (7)-(8), if the numbering is chosen as shown in (b); (b)- Optimal pyramidal BTSP tour $\langle 1, 2, 3, 4, 5, 8, 9, 12, 11, 10, 7, 6\rangle$.

This results in $O(k^3)$ time needed for each candidate at the position $\gamma(1)$ and, therefore, overall a running time of $O(k^4)$ as claimed. $\square$

To illustrate the algorithm we consider the BTSP with a rectilinear distance matrix (see Fig. 5) where the distances between points are calculated as $c_{ij} = |x_i - x_j| + |y_i - y_j|$.

$$A = \begin{pmatrix} 17 & 39 & 45 & 64 & 66 & 72 \\ 15 & 27 & 35 & 52 & 54 & 60 \\ 19 & 29 & 33 & 48 & 50 & 56 \\ 43 & 37 & 25 & 24 & 26 & 32 \\ 58 & 36 & 26 & 15 & 11 & 17 \\ 75 & 33 & 29 & 12 & 6 & 6 \end{pmatrix}$$

$$A' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & -12 & -10 & -12 & -12 & -12 \\ 2 & -10 & -12 & -16 & -16 & -16 \\ 26 & -2 & -20 & -40 & -40 & -40 \\ 41 & -3 & -19 & -49 & -55 & -55 \\ 58 & -6 & -16 & -52 & -60 & -66 \end{pmatrix}$$

The index of the maximal entries in A' in rows $2, \ldots, 6$ is 1, so $\delta(1) = 1$, i.e., column 1 remains in A at the same position.

Row 1 in A is not relevant any more to further constructions, therefore we consider a 5×6 submatrix of A to choose a row to be placed at the second position in permutation γ. This submatrix and its transformation A' are shown below.

$$A_{5\times 6} = \begin{pmatrix} 15 & 27 & 35 & 52 & 54 & 60 \\ 19 & 29 & 33 & 48 & 50 & 56 \\ 43 & 37 & 25 & 24 & 26 & 32 \\ 58 & 36 & 26 & 15 & 11 & 17 \\ 75 & 33 & 29 & 12 & 6 & 6 \end{pmatrix}$$

$$A'_{5\times 6} = \begin{pmatrix} 0 & 12 & 20 & 37 & 39 & 45 \\ 0 & 10 & 14 & 29 & 31 & 37 \\ 0 & -6 & -18 & -19 & -17 & -11 \\ 0 & -22 & -32 & -43 & -47 & -41 \\ 0 & -42 & -46 & -63 & -69 & -69 \end{pmatrix}$$

The index of the maximal entries in A' in columns $2, \ldots, 6$ corresponds to row 2 in A (row 1 in the 5×6 submatrix), so $\gamma(2) = 2$.

Proceeding in the same way we eventually obtain $\gamma = \delta = id_6$ where id_6 denotes the identity permutation on $\{1, \ldots, 6\}$ which means that the initial numbering of the cities as shown in Fig. 5 yields a distance matrix that already satisfies the conditions (7)-(9), or equivalently, conditions (4).

The optimal BTSP tour can be found by finding a shortest pyramidal tour, which is the tour $\langle 1, 2, 3, 4, 5, 8, 9, 12, 11, 10, 7, 6\rangle$.

CRediT authorship contribution statement

Vladimir G. Deineko: Investigation, Methodology, Writing – original draft, Writing – review & editing, Conceptualization. **Bettina Klinz:** Investigation, Methodology, Writing – original draft, Writing – review & editing, Conceptualization. **Gerhard J. Woeginger:** Conceptualization, Methodology, Writing – original draft.

Acknowledgements

This research has been supported by the Austrian Science Fund (FWF): W1230. The authors thank Natalia Chakhlevitch and Sigrid Knust for helpful discussions and comments on an earlier version of this paper, in particular for a helpful reference to the crane loading problem.

References

[1] D.L. Applegate, R.E. Bixby, V. Chvatal, W.J. Cook, The Traveling Salesman Problem. A Computational Study, Princeton University Press, 2006.
[2] S. Anily, R. Hassin, The swapping problem, Networks 22 (1992) 11–18.
[3] M.J. Atallah, S.R. Kosaraju, Efficient solutions to some transportation problems with applications to minimizing robot arm travel, SIAM J. Comput. 17 (1988) 419–433.
[4] A. Baltz, Algorithmic and probabilistic aspects of the bipartite traveling salesman problem, PhD Thesis, University of Kiel, 2001.
[5] A. Baltz, A. Srivastav, Approximation algorithms for the Euclidean bipartite TSP, Oper. Res. Lett. 33 (2005) 403–410.
[6] S. Bandyapadhyay, A. Banik, S. Bhore, M. Nöllenburg, Geometric planar networks on bichromatic collinear points, Theor. Comput. Sci. 895 (2021) 124–136.
[7] C. Bierwirth, F. Meisel, A survey of berth allocation and quay crane scheduling problems in container terminals, Eur. J. Oper. Res. 202 (2010) 615–627.
[8] C. Bierwirth, F. Meisel, A follow-up survey of berth allocation and quay crane scheduling, Eur. J. Oper. Res. 244 (2015) 675–689.
[9] N. Boysen, M. Fliedner, F. Jaehn, E. Pesch, A survey on container processing in railway yards, Transp. Sci. 47 (2013) 312–329.

[10] 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 (3) (1998) 496–546.
[11] R.E. Burkard, V. Deineko, On the traveling salesman problem with a relaxed Monge matrix, Inf. Process. Lett. 67 (1998) 231–237.
[12] R.E. Burkard, V. Deineko, On the Euclidean TSP with a permuted Van der Veen matrix, Inf. Process. Lett. 91 (6) (2004) 259–262.
[13] H.J. Carlo, I.F.A. Vis, K.J. Roodbergen, Storage yard operations in container terminals: literature overview, trends, and research directions, Eur. J. Oper. Res. 235 (2014) 412–430.
[14] P. Chalasani, R. Motwani, A. Rao, Approximation algorithms for robot grasp and delivery, in: Proceedings of the 2nd International Workshop on Algorithmic Foundations of Robotics, Toulouse, France, 1996, pp. 347–362.
[15] V.G. Deineko, J.A. van der Veen, R. Rudolf, G.J. Woeginger, Three easy special cases of the Euclidean travelling salesman problem, RAIRO Oper. Res. 31 (1997) 343–362.
[16] V.G. Deineko, G.J. Woeginger, The maximum traveling salesman problem on symmetric Demidenko matrices, Discrete Appl. Math. 99 (2000) 413–425.
[17] V.G. Deineko, G.J. Woeginger, Another look at the shoelace TSP: the case of very old shoes, in: Proceedings of the 7th International Conference Fun with Algorithms (FUN 2014), in: LNCS, vol. 8496, 2014, pp. 123–136.
[18] V.G. Deineko, B. Klinz, A. Tiskin, G.J. Woeginger, Four-point conditions for the TSP: the complete complexity classification, Discrete Optim. 14 (2014) 147–159.
[19] A. Frank, B. Korte, E. Triesch, J. Vygen, On the bipartite travelling salesman problem, Report No. 98866-OR, Research Institute for Discrete Mathematics, University of Bonn, 1998.
[20] A. Garcia, J. Tejel, Polynomially solvable case of the bipartite traveling salesman problem, Eur. J. Oper. Res. 257 (2017) 429–438.
[21] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman and Company, San Francisco, 1979.
[22] P.C. Gilmore, E.L. Lawler, D.B. Shmoys, Well-solved special cases, Chapter 4 in [30], 87–143.
[23] G. Gutin, A.P. Punnen, The Travelling Salesman Problem and Its Variations, Kluwer Academic Publishers, 2002.
[24] J.H. Halton, The shoelace problem, Math. Intell. 17 (4) (1995) 36–41.
[25] Y. Karuno, H. Nagamochi, A. Shurbevski, Constant factor approximatio algorithms for repetitive routing problems of grasp-and-delivery robots in production of printed circuit board, J. Oper. Res. Soc. Jpn. 55 (2001) 181–191.
[26] S.N. Kabadi, Polynomially solvable cases of the TSP, Chapter 11 in [23], 489–583.
[27] G. Kovács, Z. Tuza, B. Vizvári, H.K. Jabbari, A note on the polytope of bipartite TSP, Discrete Appl. Math. 235 (2018) 92–100.
[28] T. Leipälä, O. Nevalainen, Optimization of the movements of a component placement machine, Eur. J. Oper. Res. 38 (1989) 167–177.
[29] J. Lehnfeld, S. Knust, Loading, unloading and premarshalling of stacks in storage areas: survey and classification, Eur. J. Oper. Res. 239 (2014) 297–312.
[30] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys, The Traveling Salesman Problem, Wiley, Chichester, 1985.
[31] C. Michel, H. Schroeter, A. Srivastav, Approximation algorithms for pick-and-place robots, Ann. Oper. Res. 47 (2001) 321–338.
[32] M. Misiurewicz, Lacing irregular shoes, Math. Intell. 18 (1996) 32–34.
[33] G. Steiner, Z. Xue, The maximum traveling salesman problem on Van der Veen matrices, Discrete Appl. Math. 146 (2005) 1–2.
[34] J.A.A. van der Veen, A new class of pyramidally solvable symmetric traveling salesman problems, SIAM J. Discrete Math. 7 (1994) 585–592.