A new asymmetric pyramidally solvable class of the traveling salesman problem

Mohammed Fazle Baki

Odette School of Business, University of Windsor, Windsor, Ont., Canada, N9B 3P4

Received 31 October 2004; accepted 14 December 2005

Abstract

We present a new method of identifying a class of asymmetric matrices for which an optimal traveling salesman tour exists that is pyramidal. The new class generalizes two previously known classes of matrices and includes some new matrices as well.

MSC: 90C27

Keywords: Traveling salesman problem; Pyramidal tour; Polynomial time algorithm

1. Introduction

Given a complete directed graph \( G \) on node set \( N = \{1, 2, \ldots, n\} \) and an \((n \times n)\) cost matrix, \( C \) such that the cost of traversing arc \((i, j)\) is \( C(i, j) \) (we assume throughout that \( C(i, i) = 0 \forall i \)), the traveling salesman problem (TSP) is to find a tour \( \tau \) such that \( C(\tau) = \sum C(i, j) : (i, j) \in \tau \) is minimum.

Since TSP is NP-hard [13], researchers have identified several polynomially solvable cases of it. For a survey on such cases, we refer the reader to [5,15,16,18,22]. Among the well-studied solvable classes is the one for which an optimal tour exists which is pyramidal (i.e., of the form \((1 - i_1, -i_2 - \cdots - i_p - n - j_1 - j_2 - \cdots - j_s - 1)\), where, \( i_1 < i_2 < \cdots < i_p \) and \( j_1 > j_2 > \cdots > j_s \)). The significance of this case lies in the fact that for a given ordering of the nodes, (i) the total number of pyramidal tours is exponentially large and (ii) an optimal pyramidal tour can be computed in \( O(n^2) \) time [5,15,18,21]. It is shown in [25] that an optimal pyramidal tour can be computed in \( O(n) \) time if the cost matrix satisfies what is known as “the Monge condition” [5]. The notion of pyramidal tours has been generalized to pyramidal tours with step-back peaks in [12,23,24]. The class of pyramidal tours with step-back peaks is solvable in \( O(n^2) \) time [12,23,24].

Since the class of pyramidal tours is exponentially large and since an optimal pyramidal tour can be found in polynomial time, the class of pyramidal tours has drawn the attention of many researchers in the context of a neighborhood search.
In Section 3, we discuss a new pyramidally solvable class of the TSP. This method incorporates the algorithm for finding an optimal pyramidal path into the algorithm time, if a matrix belongs to the new class. In Section 4, we discuss its relationship to other known classes of pyramidally solvable traveling salesman problems.

2. Notation and definitions

A path $\mu$ from a node $u$ to a node $v$, denoted by $[u, v]$, is a sequence of arcs of the type $(k_0, k_1) − (k_1, k_2) − \cdots − (k_m, k_{m+1})$ where $k_0 = u$ and $k_{m+1} = v$. We shall refer to such a path by $(k_0 − k_1 − \cdots − k_m − k_{m+1})$. In this case, $\mu^{-1}$ is the path $(k_{m+1} − k_m − \cdots − k_1 − k_0)$. For any $i < (m + 1)$ and $j > 0$, we define $\mu(k_i) = k_{i+1}$ and $\mu^{-1}(k_j) = k_{j-1}$. Path $\mu$ is simple if all the nodes $\{k_i: i = 0, 1, \ldots, (m+1)\}$ are distinct, except for, possibly, the end nodes. If $u = v$, then path $\mu$ is closed. Path $\mu$ is a Hamiltonian path on node set $N'$ in $G$ if $\mu$ is simple and $N' = \{k_i: i = 0, 1, \ldots, (m+1)\}$. A tour $\tau$ is a closed Hamiltonian path on node set $N$.

A peak of a path $\mu$ is a node $p$ in $\mu$ such that (i) $p > \mu(p)$ (if $\mu(p)$ exists) and (ii) $p > \mu^{-1}(p)$ (if $\mu^{-1}(p)$ exists) and a valley of $\mu$ is a node $q$ in $\mu$ such that (i) $q < \mu(q)$ (if $\mu(q)$ exists) and (ii) $q < \mu^{-1}(q)$ (if $\mu^{-1}(q)$ exists). A pyramidial path is a path with only one peak and/or one valley.

For an $n \times n$ cost matrix $C$, $C^T$ is the transpose of $C$, and $C_R$ is the reverse of $C$, i.e., $\forall 1 \leq i, j \leq n$, $C_R(i, j) = C(n − i + 1, n − j + 1)$. The cost $C(\mu)$ of a path $\mu$ corresponding to cost matrix $C$ is the sum of costs of the arcs in $\mu$.

Definition 1. For any tour $\tau$ on node set $\{1, 2, \ldots, n\}$ and any $i \in \{1, 2, \ldots, n\}$, $\tau^{(i)}$ is a tour on node set $\{1, 2, \ldots, i\}$ defined recursively as: $\tau^{(n)} = \tau$, and for $i = (n − 1), (n − 2), \ldots, 2$, tour $\tau^{(i)}$ is obtained from $\tau^{(i+1)}$ by replacing subpath $(u − (i + 1) − v)$ of $\tau^{(i+1)}$ with arc $(u, v)$.

3. A new pyramidally solvable class of the TSP

As can be seen in Appendix A, most previously known classes of pyramidally solvable TSPs have been described in terms of linear inequalities, which involve entries of the underlying cost matrix. In this paper, we present a new method of identifying pyramidally solvable TSPs. This method incorporates the algorithm for finding an optimal pyramidal path into the algorithm.

[26,27] and the development of heuristics [6]. See [7,8,17] for many recent developments on exponential neighborhoods.

In some examples of the TSP, an optimal tour exists which is pyramidal. During the last three decades, a great deal of work has been done on the identification of polynomially testable sufficiency conditions for instances where a TSP is thought to be pyramidally solvable. “The Demidenko condition” [10] generalizes many conditions developed previously. The class of Demidenko matrices drew the attention of many researchers. Its proof, generalization and recognition have been discussed in [3,5,15,18,25,31]. The first condition for pyramidal solvability that differs from the Demidenko condition is the van der Veen condition [30], which is applicable to the symmetric TSP. An asymmetric analog of the van der Veen condition has been given in [23,24]. While each Demidenko condition and van der Veen condition guarantees a minimum cost traveling salesman tour in polynomial time, finding a maximum cost traveling salesman tour is NP-hard on each of the symmetric Demidenko matrices [9] and van der Veen matrices [28]. The Demidenko condition, van der Veen condition and the most general conditions for pyramidal solvability are given in Appendix A. See [1–3,5,10–12,15,18,23,24,29,30,32,33] for details.

Checking if an optimal tour exists that is pyramidial is important in the context of several solvable cases, including those for which the algorithm of Gilmore–Gomory [14] guarantees an optimal solution. A large class [4,5,15,18–20] of TSPs is solvable by the algorithm of Gilmore–Gomory. The problem of checking if an optimal tour exists that is pyramidial arises [5,20,18] as a subproblem when checking if an instance of TSP is solvable by the algorithm of Gilmore–Gomory.

Every pyramidally solvable case of the TSP included in Appendix A satisfies a hereditary property [3]. If a TSP satisfies a hereditary property $P$, then every TSP obtained by removing nodes $1, \ldots, i$ and/or $j, \ldots, n$ for any $1 \leq i < j \leq n$ satisfies property $P$. The characterization of symmetric or asymmetric matrices that satisfy a hereditary property sufficient for pyramidal solvability is an open problem.

In Section 2, we introduce notation and definitions. In Section 3, we discuss a new pyramidally solvable class of TSP, showing that we can verify, in polynomial time, if a matrix belongs to the new class. In Section 4, we discuss its relationship to other known classes of pyramidally solvable traveling salesman problems.
Theorem 1. If, for every pyramidal Hamiltonian path \( \mu = [a, b] \) on node set \( \{a, b\} \cup \{p + 1, p + 2, \ldots, x\} \), \( 3 \leq p < x \leq n, 1 \leq a, b < p, a \neq b \) and every \( \{u, v\} \equiv \{p - 1, p\} \), there exists a pyramidal Hamiltonian path \( \mu' = [u, v] \) on node set \( \{p - 1, p, \ldots, x\} \) such that \( C(\mu') - C(\mu) \leq C(\mu) - C(a, b) \), then an optimal tour exists which is pyramidal.

Proof. This claim can be verified when \( n = 4 \). Suppose also that the claim is not true in some cases where \( n > 4 \). Choose the lowest value of \( n \), such that an \((n \times n)\) matrix \( C \) exists which satisfies the conditions of the theorem and for which no optimal tour exists that is pyramidal. Let \( \tau \) be an optimal tour and \( p \) be its largest peak less than \( n \). Then \( \tau \) contains a subpath \( \mu = [a, b], a < p, b < p, a \neq b \) such that \( \mu \) is pyramidal and Hamiltonian on node set \( \{a, b\} \cup \{p + 1, p + 2, \ldots, n\} \). Identify an optimal pyramidal tour \( \tau' \) on node set \( N \{p + 1, p + 2, \ldots, n\} \). By the choice of \( n \), \( C(\tau') \leq C(\tau^{(p)}) \). Let arc \((u, v) \in \tau'\), \( \{u, v\} \equiv \{p - 1, p\} \) (since \( \tau' \) is pyramidal, one such arc must exist). Construct a pyramidal tour \( \tau^* \) from \( \tau' \) by replacing arc \((u, v) \) with subpath \( \mu' = [u, v] \) such that \( \mu' \) is pyramidal and Hamiltonian on node set \( \{p - 1, p, \ldots, n\} \) and \( C(\mu') - C(u, v) \leq C(\mu) - C(a, b) \). Now, \( C(\tau^*) = C(\tau') - C(u, v) + C(\mu') \leq C(\tau^{(p)}) - C(a, b) + C(\mu) = C(\tau) \). Contradiction. \( \square \)

Given a cost matrix, one may like to check whether the conditions of Theorem 1 are satisfied. It is sufficient to check if, for every 4-tuple of integers \( a, b, p, x \), the cost of replacing arc \((a, b) \) with an optimal pyramidal path \([a, b]\) is not less than the cost of replacing any arc of the type \((u, v), \{u, v\} \equiv \{p - 1, p\} \) with an optimal pyramidal path \([u, v] \). To be more precise, let \( \mu_{i, j, k, l} \) be an optimal pyramidal path of the type \( [i, j] \) and Hamiltonian on \( [i, j] \cup \{k, k + 1, \ldots, l\} \). It is sufficient to check if, \( \forall 1 \leq a \neq b < p < x \leq n, \)

\[
C(\mu_{a, b, p + 1, x}) - C(a, b) \leq \max \left\{ \begin{array}{l}
C(\mu_{p - 1, p + 1, x}) - C(p - 1, p), \\
C(\mu_{p - 1, p, p + 1, x}) - C(p, p - 1)
\end{array} \right\}.
\]

Since \( O(n^2) \) time is required to compute an optimal pyramidal path, a straightforward computation would lead to an \( O(n^6) \) algorithm that tests all conditions of \( a, b, p, x \). However, the computational complexity of the recognition algorithm can be improved. For this, we need to introduce some notation. For any \( 1 \leq i < k \leq n \), let

\[
\begin{align*}
&f_i,k = \text{Cost of an optimal pyramidal Hamiltonian path } [i + 1, i] \text{ on } [i, i + 1, \ldots, k] \\
g_i,k = \text{Cost of an optimal pyramidal Hamiltonian path } [i, i + 1] \text{ on } [i, i + 1, \ldots, k]
\end{align*}
\]

If \( x = p + 1 \), then

\[
C(\mu_{a, b, p + 1, x}) = C(a, p + 1) + C(p + 1, b).
\]

If \( x \geq p + 2 \), then

\[
C(\mu_{a, b, p + 1, x}) = \min_{p + 2 \leq j \leq x} \left\{ \begin{array}{l}
C(a, j) + C(p + 1, b) + h^{'1, p + 1, j - 1} \\
+ C(\mu_{j, j - 1, j + 1, x}) + C(a, p + 1) + C(j, b) + h_{p + 1, j - 1} \\
+ C(\mu_{j - 1, j - 1, j, x})
\end{array} \right\}.
\]

Now, knowing that \( C(\mu_{j, j - 1, j + 1, x}) \) = \( f_j - 1, x \), \( C(\mu_{j - 1, j - 1, j, x}) = g_{j - 1, x} \), \( C(\mu_{p - 1, p, p + 1, x}) = f_p - 1, x \), \( C(\mu_{p - 1, p, p + 1, x}) = g_{p - 1, x} \) and rearranging terms, we can rewrite the conditions of Theorem 1 as follows: 

\[
H(p, x) \geq \max \left\{ \begin{array}{l}
g_{p - 1, x} - C(p - 1, p), \\
g_{p - 1, x} - C(p, p - 1)
\end{array} \right\},
\]

where

\[
H(p, x) = \left\{ \begin{array}{ll}
\min_{1 \leq a \neq b < p} \{C(a, p + 1) \\
+ C(p + 1, b) - C(a, b)\} & \text{iff } x = p + 1, \\
\min_{p + 2 \leq j \leq x} \{F(p, j) + f_{j - 1, x}\} & \text{iff } x \geq p + 2,
\end{array} \right.
\]

where \( F(p, j) = \min_{1 \leq a \neq b < p} \{C(a, j) + C(p + 1, b) - C(a, b)\} + h^{'p + 1, j - 1} \).
\[ G(p, j) = \min_{1 \leq a < b < p} \{ C(a, p+1) + C(j, b) - C(a, b) \} + h_{p+1, j-1}. \]

**Theorem 2.** The conditions of Theorem 1 can be verified in \( O(n^4) \) time and \( O(n^2) \) space.

**Proof.** The conditions of Theorem 1 are equivalent to the following: \( \forall 3 \leq p < x \leq n, \)
\[
H(p, x) = \max \left\{ g_{p-1, x} - C(p, 1, p), f_{p-1, x} - C(p, p-1) \right\}.
\]

If \( x = p + 1, \)
\[
H(p, p + 1) = \min_{1 \leq a < b < p} \{ C(a, p + 1) + C(p + 1, b) - C(a, b) \},
\]
and if \( x \geq p + 2, \)
\[
H(p, x) = \min_{p+2 \leq j \leq x} \left\{ f_{j-1, x} + F(p, j), g_{j-1, x} + G(p, j) \right\},
\]
\[
\begin{align*}
g_{p-1, x} &= \min_{p+2 \leq j \leq x} \left\{ f_{j-1, x} + C(p, 1) + C(p+1, p) + h_{p+1, j-1}' \right\}, \\
f_{p-1, x} &= \min_{p+2 \leq j \leq x} \left\{ g_{j-1, x} + C(p, 1) + C(p+1, p) + h_{p+1, j-1}' \right\}.
\end{align*}
\]

The pre-computation of \( h, h', F \) and \( G \) requires \( O(n^4) \) time and \( O(n^2) \) space. For each \( p \), the case, when \( x = p + 1 \), can be verified in \( O(n^2) \) time and each case, when \( x \geq p + 2 \), can be verified in \( O(n) \) time. Since \( O(n) \) cases exist when \( x \geq p + 2 \), the conditions can be verified in \( O(n^2) \) time for each \( p \). Thus, all conditions can be verified in \( O(n^3) \) time. The bottleneck occurs in the pre-computation of \( F \) and \( G \), which requires \( O(n^4) \) time. \( \square \)

4. **Examples and relationships with other pyramidally solvable classes**

In this section, we shall show that the conditions of Theorem 1 generalize some previously known classes of pyramidally solvable TSPs and include some new matrices.

If a cost matrix satisfies Baki–Kabadi conditions [2] I, II, or the van der Veen condition [30], then the matrix satisfies the conditions of Theorem 1. For example, consider a pyramidally Hamiltonian path \( \mu = [a, b] \) on node set \( \{a, b\} \cup \{p + 1, p + 2, \ldots, x\} \), where \( 1 \leq a, b \neq p < x < n \). To show that the conditions of Theorem 1 hold, we shall construct pyramid Hamiltonian paths \( \mu_1 = [p - 1, p] \) and \( \mu_2 = [p, p - 1] \) on node set \( \{p - 1, p, \ldots, x\} \), such that
\[
C(\mu) - C(a, b) \geq C(\mu_1) - C(p - 1, p) \quad \text{and} \quad C(\mu) - C(a, b) \geq C(\mu_2) - C(p, p - 1).
\]

Suppose that \( a < b \). The case with \( b < a \) is similar. Let arcs \( (a, k_1) \in \mu \) and \( (k_2, b) \in \mu \). Construct path \( \mu_1 \) from \( \mu \) by replacing arc \( (a, k_1) \) with \( (p - 1, k_1) \) and \( (k_2, b) \) with \( (k_2, p) \). Construct path \( \mu_2 \) from \( \mu \) by replacing arc \( (a, k_1) \) with \( (p, k_1) \) and \( (k_2, b) \) with \( (k_2, p - 1) \). If a cost matrix satisfies Baki–Kabadi conditions I or II, \( \mu_1 = \mu_1 \) and \( \mu_2 = \mu_2 \). If a cost matrix satisfies the van der Veen condition and \( (p - b) \) is even, then \( \mu_1 = \mu_1 \) and \( \mu_2 = \mu_2 \); if \( (p - b) \) is odd, then \( \mu_1 = \mu_1 \) and \( \mu_2 = \mu_2 \).

The conditions of Theorem 1 differ from the conditions known previously and listed in Appendix A. To show this, we shall give an example of an asymmetric matrix and another example of a symmetric matrix.

**Example 1.** Consider a TSP with the following cost matrix:
\[
B_1 = \begin{pmatrix}
0 & 18 & 13 & 17 & 32 & 11 \\
13 & 0 & 11 & 15 & 13 & 28 \\
12 & 17 & 0 & 16 & 14 & 22 \\
22 & 25 & 20 & 0 & 22 & 16 \\
18 & 21 & 14 & 17 & 0 & 11 \\
16 & 19 & 13 & 15 & 15 & 0
\end{pmatrix}.
\]
Example 2. Consider a TSP with the following cost matrix:

\[
B_2 = \begin{pmatrix}
0 & 9 & 6 & 22 & 6 & 54 \\
9 & 0 & 15 & 13 & 9 & 46 \\
6 & 15 & 0 & 28 & 10 & 61 \\
22 & 13 & 28 & 0 & 20 & 32 \\
6 & 9 & 10 & 20 & 0 & 52 \\
54 & 46 & 61 & 32 & 52 & 0
\end{pmatrix}.
\]

Both matrices \(B_1\) and \(B_2\) satisfy the conditions of Theorem 1. However, as shown in Appendix A, the matrices do not satisfy any previously known condition sufficient for the pyramidal solvability of TSP.

Acknowledgement

This research has been partially supported by the Natural Sciences and Engineering Research Council (NSERC). Thanks are due to Dr. S.N. Kabadi for a fruitful discussion and for his encouragement in publishing the results. Thanks are due also to an anonymous referee for his/her helpful comments and constructive suggestions.

Appendix A

For any \((n \times n)\) matrix \(A\), let

\[
R_A(i, j, k, l) = A(i, l) + A(k, j) - A(i, j) - A(k, l)
\]

\(\forall 1 \leq i < k \leq n, 1 \leq j < l \leq n,\)

\[
D_A(i, j) = R_A(i, j, i + 1, j + 1) \quad \forall 1 \leq i, j \leq n - 1,
\]

\[
I_A(i, j, k) = R_A(i, j, k, j + 1) \quad \forall 1 \leq i < j < k \leq n,
\]

\[
L_A(i, j, k) = R_A(i, j, j + 1, j + 1) + R_A(j, j + 1, k + 1, k) \quad \forall 1 \leq i < j + 1 < k \leq n,
\]

\[
V_A(i, j, k) = R_A(i, j, j + 1, k) \quad \forall 1 \leq i < j + 1 < k \leq n,
\]

\[
GW_A(a, b, x, u, v) = A^0(\tau_1) - A^0(\tau_2),
\]

\(\forall 1 \leq a < b < x < 1 < n, 1 \leq u, v < x - 1, u \neq v, u \neq a, v \neq b,\)

where \(A^0\) is the \(4 \times 4\) submatrix of \(A\), with rows indexed by \(a, u, (x - 1), x\) (in that order), then \(\tau_1\) is an optimal non-pyramidal tour and \(\tau_2\) is an optimal pyramidal tour, each for a TSP on node set \(\{1, 2, 3, 4\}\) with a cost matrix of \(A^0\).

\[
GD_A(a, b, x) = d_{a,b}(x - 1) + \max \left( \sum_{k=b+1}^{x-1} R(k, k-1, x, k), \sum_{k=b+1}^{x-1} R(a, k, k, k+1) \right),
\]

\(\forall 1 \leq a < b < x - 1 < n,\)

where \(d_{a,b}(x - 1)\) is the length of the shortest path from node \(b\) to node \((x - 1)\) in a network, with node set \(N = \{b, b + 1, \ldots, (x - 1)\}\), arc set \(E = \{(i, j): b \leq i < j \leq (x - 1)\}\) and arc weight function \(r_{i,j}\) defined as

\[
r_{b,(x-1)} = \min\{S(u, v, b+1, x-1): u, v \leq b; u \neq v; u \neq a; v \neq b; and if a \neq 1,\ then v \neq a\},
\]

\[
r_{b,j} = \min\{S(u, v, b+1, j): u, v \leq b; u \neq v; u \neq a; v \neq b\} \quad \forall b < j < (x - 1),
\]

\[
r_{i,j} = \min\{S(u, v, i + 1, j): u, v \leq (i - 2); u \neq v; u \neq a; v \neq b\} \quad \forall b < i < j \leq (x - 1),
\]

\[
S(u, v, i, j) = \left\{ \begin{array}{l}
R(u, v, i, j) \\
+ \min \left( \sum_{k=i+1}^{j} R(u, k - 1, k, k), \sum_{k=i+1}^{j} R(k - 1, v, k, k) \right) \\
\end{array} \right. 
\]

\[
= \min \left\{ \sum_{k=i}^{j} R(i - 2, k - 1, k, k), \sum_{k=i}^{j} R(k - 1, i - 2, k, k) \right. 
\]

The suffix \(A\) is omitted whenever there is no ambiguity.
Demidenko condition [10]. Both $C$ and $C^T$ satisfy the following:

For any $1 \leq a < b < b + 1 < x \leq n$,

(i) $I(a, b, x) \geq 0$ and

(ii) $L(a, b, x) \geq 0$.

Demidenko condition [11]. Matrix $C$ satisfies the following:

(i) $D(a, b) \geq 0$ for all $1 \leq a < b < n$, $|a - b| > 1$,

(ii) $R(a, a + 1, a + 3, a + 2) \geq 0$ for all even $a$ with $2 \leq a \leq n - 3$,

(iii) $R(a + 1, a, a + 2, a + 3) \geq 0$ for all odd $a$ with $1 \leq a \leq n - 3$,

(iv) $R(a + 1, a + 1, a + 2, a + 2) \geq R(a + 1, a, a + 2, a + 1)$ for all even $a$ with $2 \leq a \leq n - 2$,

(v) $R(a + 1, a, a + 2, a + 1) \geq R(a + 1, a + 1, a + 1, a + 2)$ for all odd $a$ with $1 \leq a \leq n - 2$.

Van der Veen condition [30]. Cost matrix $C$ is symmetric and for any $1 \leq a < b < b + 1 < x \leq n$,

(i) $V(a, b, x) \geq 0$.

Baki–Kabadi condition I [2]. Either (i) both $C$ and $C^T$ or (ii) both $C_R$ and $C_R^T$ satisfy the following: for any $1 \leq a < b < b + 1 < u, v \leq n$,

(i) $R(a, b, b, u) + R(b, b, v, b + 1) \geq 0$ and

(ii) $I(a, b, u) \geq 0$.

Baki–Kabadi condition II [2]. Either (i) both $C$ and $C^T$ or (ii) both $C_R$ and $C_R^T$ satisfy the following: for any $1 \leq a < b < b + 1 < u, v \leq n$,

(i) $R(a, b, b, u) + R(b, b, v, b + 1) \geq 0$ and

(ii) $V(a, b, u) \geq 0$.

Enomoto–Oda–Ota condition [12]. Both $C$ and $C^T$ satisfy the following: for any $1 \leq a < b < b + 1 < x \leq n$,

(i) $I(a, b, x) \geq 0$ and

(ii) $V(a, b, x) \geq 0$.

Oda condition I [23]. Both $C$ and $C^T$ satisfy the following: for any $1 \leq a < b < u < v \leq n$,

(i) $R(a, b, b, v) + R(b, b, v, u) \geq 0$,

(ii) $R(u, a, v, u) + R(a, b, u, u) \geq 0$ and

(iii) $R(a, b, u, v) \geq 0$.

Oda condition II [24]. Both $C$ and $C^T$ satisfy the following: for any $1 \leq a < b < u < v \leq n$,

(i) $R(a, b, u, v) \geq 0$.

Generalized Demidenko–Warren condition [3]. Either $\forall 1 \leq a < b < x - 1 < n$ in each of $C$ and $C^T$ or, $\forall 1 \leq a < b < x - 1 < n$ in each of $C_R$ and $C_R^T$, satisfies the following:

(i) $GW(a, b, x, u) \geq 0 \forall 1 \leq u, v < x - 1, u \neq v, u \neq v, u \neq a, v \neq b$ or

(ii) $GD(a, b, x) \geq 0$.

Claim 1. Matrix $B_1$ does not satisfy any condition listed in Appendix A.

Proof. Since the matrix is asymmetric, and since $D(1, 5) = -36 < 0$, $I(3, 4, 6) = -2 < 0$, $L(3, 4, 6) = -2 < 0$, and $V(1, 2, 6) = -12 < 0$, the matrix does not satisfy the Demidenko [10,11], van der Veen [30], Baki–Kabadi [2] I and II, Enomoto–Oda–Ota [12], Oda I [23] and Oda II [24] conditions. Note that the Oda I [23] condition (iii) and the Oda II [24] condition (i) $R(a, b, u, v) \geq 0$ reduce to $V(a, b, v) \geq 0 \forall 1 \leq a < b < b + 1 = u < v \leq n$. To check if matrix $B_1$ satisfies the generalized Demidenko–Warren [3] condition, consider $a = 1$, $b = 4$, and $x = 4$. Let $u = 2$ and $v = 3$. To calculate $GW_{B_1}(1, 4, 6, 2, 3)$, let $B_{1}^{0}$ be the $4 \times 4$ submatrix of $B_1$ with rows 1, 2, 5, 6 and columns 3, 4, 5, 6:

$$B_{1}^{0} = \begin{pmatrix}
13 & 17 & 32 & 11 \\
11 & 15 & 13 & 28 \\
14 & 17 & 0 & 11 \\
13 & 15 & 15 & 0
\end{pmatrix}.$$

For TSPs with a cost matrix of $B_{1}^{0}$, the cost of an optimal non-pyramidal tour is 53 and the cost of an optimal pyramidal tour is 54. Hence, $GW_{B_1}(1, 4, 6, 2, 3) = 53 - 54 = -1 < 0$. To calculate $GD_{B_1}(1, 4, 6)$, find $d_{1, 4}(5) = r_{4, 5} = \min\{S(u, v, 5, 5) := u \neq v < 4, u \neq 1, v \neq 4\} = R(3, 4, 5, 5) = 15$ and $\max\{R(5, 4, 6, 5), R(1, 5, 5, 6)\} = -17$. Hence, $GD_{B_1}(1, 4, 6) = 15 - 17 = -2 < 0$. Next, consider matrix $B_3(B_1)_R$. Consider also $a = 1$, $b = 4$, and $x = 6$. Let $u = 4$ and $v = 1$. To calculate $GW_{B_1}(1, 4, 6, 4, 1)$, let $B_{3}^{0}$ be the $4 \times 4$ submatrix of $B_3$ with rows and
Claim 2. Matrix $B_2$ does not satisfy any condition listed in Appendix A.

Proof. Since $D(1, 4) = -12 < 0$, $I(1, 2, 4) = -18 < 0$, $L(1, 2, 6) = -18 < 0$, and $V(1, 2, 6) = -1 < 0$, the matrix does not satisfy the Demidenko–Warren [10,11], van der Veen [30], Baki–Kabadi [2] I and II, Enomoto–Oda–Ota [12], Oda I [23] and Oda II [24] conditions. To check if matrix $B_2$ satisfies the generalized Demidenko–Warren [3] condition, consider $a = 2$, $b = 4$, and $x = 6$. Let $u = 3$ and $v = 2$. To calculate $GW_{B_2}(2, 4, 6, 3, 2)$, let $B_2^0$ be the $4 \times 4$ submatrix of $B_2$ with rows 2, 3, 5, 6 and columns 2, 4, 5, 6:

$$B_2^0 = \begin{pmatrix} 0 & 13 & 9 & 46 \\ 15 & 28 & 10 & 61 \\ 9 & 20 & 0 & 52 \\ 46 & 32 & 52 & 0 \end{pmatrix}.$$

For TSPs with a cost matrix of $B_2^0$, the cost of an optimal non-pyramidal tour is 97 and the cost of an optimal pyramidal tour is 108. Hence, $GW_{B_2}(2, 4, 6, 3, 2) = 97 - 108 = -11 < 0$. To calculate $GD_{B_2}(2, 4, 6)$, find $d_{2,4}(5) = r_{4,5} = \min\{S(u, v, 5, 5) : u \neq v \leq 4, u \neq 2, v \neq 4\} = R(3, 4, 5, 5) = 2$ and $\max\{R(5, 4, 6, 5), R(2, 5, 6, 6)\} = -15$. Hence, $GD_{B_2}(2, 4, 6) = 2 - 15 = -13 < 0$. Next, consider matrix $B_2^0 \oplus B_2$. Consider also $a = 2$, $b = 4$, and $x = 6$. Let $u = 3$ and $v = 2$. To calculate $GW_{B_2}(2, 4, 6, 3, 2)$, let $B_2^0$ be the $4 \times 4$ submatrix of $B_2$ with rows 2, 3, 5, 6 and columns 2, 4, 5, 6:

$$B_2^0 = \begin{pmatrix} 0 & 10 & 9 & 6 \\ 20 & 28 & 13 & 22 \\ 9 & 15 & 0 & 9 \\ 6 & 6 & 9 & 0 \end{pmatrix}.$$

For TSPs with a cost matrix of $B_2^0$, the cost of an optimal non-pyramidal tour is 34 and the cost of an optimal pyramidal tour is 38. Hence, $GW_{B_2}(2, 4, 6, 3, 2) = 34 - 38 = -4 < 0$. To calculate $GD_{B_2}(2, 4, 6)$, find $d_{2,4}(5) = r_{4,5} = \min\{S(u, v, 5, 5) : u \neq v \leq 4, u \neq 2, v \neq 4\} = R(3, 4, 5, 5) = 0$ and $\max\{R(5, 4, 6, 5), R(2, 5, 6, 6)\} = -12$. Hence, $GD_{B_2}(2, 4, 6) = 0 - 12 = -12 < 0$. Now, since $GW_{B_2}(2, 4, 6, 3, 2) < 0$, $GD_{B_2}(2, 4, 6) < 0$, $GW_{B_2}(2, 4, 6, 3, 2) < 0$ and $GD_{B_2}(2, 4, 6) < 0$, matrix $B_2$ does not satisfy the generalized Demidenko–Warren [3] condition.

References