# Consistency of expert-based preference matrices

Martin Gavalec[a], Karel Mls[a]

[a] *University of Hradec Králové, Faculty of Informatics and Management, Rokitanského 62, 50003 Hradec Králové, Czech Republic*

*Keywords:* preference matrix, AHP decision making, cyclic consistency, consistent matrix
*2000 MSC:* 90C15

## 1. Cyclic inconsistency of a preference matrix

In AHP approach to multi-criteria decision problem, the relative importance of alternatives is computed from preference matrices, which come from experience and can possibly be inconsistent.

An algorithm for computing a consistent approximation of a given preference matrix by digraph method is described in this paper. We start with an analysis of the inconsistency of a given preference matrix. The first type of inconsistency is caused by so-called inconsistency cycles. The inconsistency of this type is removed by computing the strongly connected components in the associated digraph and a small modification. If the modified matrix is cyclic consistent, i.e. it contains no inconsistent cycles, or if some of the entries of the matrix are missing, then a consistent approximation is computed. The computational complexity of the algorithm is $O(n^2)$.

Preference matrix $A$ is called *cyclic inconsistent*, if there is a cycle

$$i_1, i_2, \ldots, i_r, i_{r+1} = i_1$$

of length $r \geq 2$ of indices in $N$ (called: inconsistent cycle in $A$) such that the inequalities

$$a(i_k i_{k+1}) \geq 1 \quad \text{for every} \quad k = 1, 2, \ldots, r \tag{1}$$

*Email addresses:* `martin.gavalec@uhk.cz` (Martin Gavalec), `karel.mls@uhk.cz` (Karel Mls)

hold, and at least one of the inequalities is strict. Matrix $A$ is cyclic consistent, if every cycle in $A$ is consistent, i.e. $A$ contains no inconsistent cycles. Examples of inconsistent cycles of length $r = 3$ are shown in Figure 1.

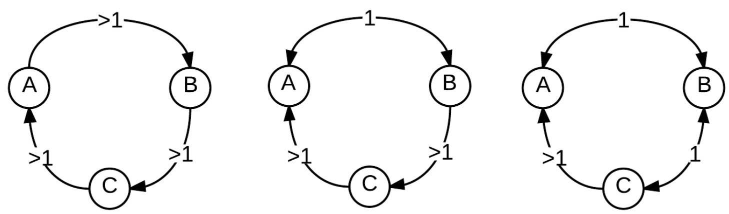


Figure 1: Possible types of cyclic inconsistency with $r = 3$

**Theorem 1.1.** *If a complete preference matrix $A$ contains an inconsistent cycle of length $r > 3$, then $A$ also contains an inconsistent cycle of length $3$.*

**Corollary 1.2.** *The cyclic consistency of a complete preference matrix can be recognized in time $O(n^3)$, by verifying all index cycles of length $3$ for the inconsistency.*

Theorem 1.1 is illustrated by Figure 2 below.

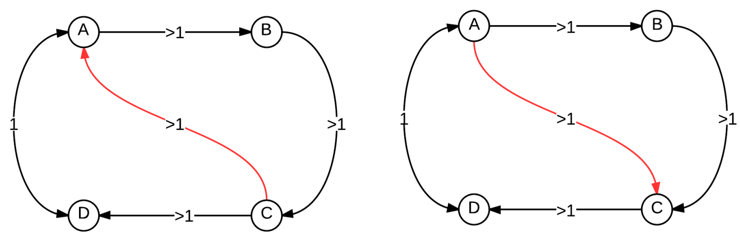


Figure 2: Cyclic inconsistency with $r = 4$

The *preference digraph* $\mathcal{D} = \big(V(A), E(A)\big)$ of a given preference matrix $A$ is defined as follows

$$\begin{aligned} V(\mathcal{D}) &= \big\{1, 2, \ldots, n\big\} \\ E(\mathcal{D}) &= \big\{\,(i,j);\, a_{ij} \geq 1\,\big\} \\ E^{+}(\mathcal{D}) &= \big\{\,(i,j);\, a_{ij} > 1\,\big\} \end{aligned}$$

The edges in $E^{+}(\mathcal{D})$ are called strong preference edges.

**Theorem 1.3.** *Preference matrix $A$ is cyclic consistent if and only if every cycle $C$ in $\mathcal{D}(A)$ contains no strong preference edges, i.e. $C \cap E^+(\mathcal{D}) = \emptyset$.*

**Theorem 1.4.** *Preference matrix $A$ is cyclic consistent if and only if every strongly connected component $\mathcal{K}$ in $\mathcal{D}(A)$ contains no strong preference edges, i.e. $(\mathcal{K} \times \mathcal{K}) \cap E^+(\mathcal{D}) = \emptyset$.*

A possible treating method (3-cycle method, for short) is based on Theorem 1.1.

**3-cycle method**

1. find all inconsistent cycles of length 3 in $A$
2. change all preferences in the cycles to 1
3. repeat 1 - 2 until there is no inconsistent cycle of length 3 in $A$

The repetition of steps 1 and 2 in the 3-cycle method is necessary , because new inconsistent cycles can be created by treating the cyclic inconsistency by cycles of length 3 (see Figure 3).

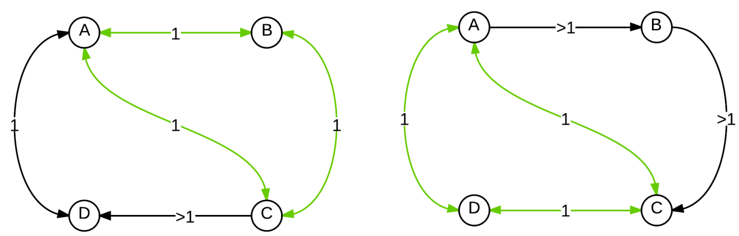


Figure 3: Creating new inconsistent cycles of lenth $r = 3$

The above disadvantage is not present at another method for treating the inconsistency (SCC method, for short) which works with strongly connected components in the preference digraph $\mathcal{D}(A)$. The method is based on Theorem 1.4.

**SCC method**

1. find all strongly connected components in digraf $\mathcal{D}(A)$
2. change all preferences within the strongly connected components to 1

**Theorem 1.5.** *If matrix $A'$ is created from preference matrix $A$ by SCC method, i.e.*

$$a'_{ij} = \begin{cases} 1 & \text{for } i,j \in \mathcal{K} \text{ in every strongly connected component } \mathcal{K} \text{ of } \mathcal{D}(A) \\ a_{ij} & \text{otherwise} \end{cases},$$

*then $A'$ is reciprocal and cyclic consistent.*

The cyclic consistent matrix $A'$ computed by the SCC algorithm is called the *cyclic consistent approximation* of $A$.

## 2. Computing consistent preferences

*Preference order* $\mathcal{P}(A)$ induced by $A$ is defined as follows: if inequalities $a(i_k i_{k+1}) \geq 1$ with $k = 1, 2, \ldots, r-1$ hold for some sequence $i = i_1, i_2, \ldots, i_r = j$, then $(i,j) \in \mathcal{P}(A)$.

**Theorem 2.1.** *If a preference matrix $A$ is cyclic consistent, then $\mathcal{P}(A)$ is a uniquely determined linear order of alternatives (up to permutations of equivalent preferences).*

**Theorem 2.2.** *If $A$ is a preference matrix and $A'$ is its cyclic consistent approximation, then the linear order $\mathcal{P}(A')$ of alternatives is equal to the order of strictly connected components in preference digraph $\mathcal{D}(A)$.*

**Algorithm ConsistApprox**

[1] Input: preference matrix $A$
[2] compute preference digraph $\mathcal{D}(A)$
[3] compute cyclic consistent matrix $A'$ by SCC method
[4] compute linear order $\mathcal{P}(A')$ induced by the order of strongly connected components in digraph $\mathcal{D}(A)$
[5] for any component $\mathcal{K}$ and its successor $\mathcal{L}$ substitute values ${a'}_{ij}$ with $i \in \mathcal{K},\ j \in \mathcal{L}$ by their common geometric mean $\tilde{a}_{ij}$
[6] extend the 'overdiagonal block' values in matrix $\tilde{A}$ using the reciprocity and consistency condition
[7] Output: reciprocal and consistent approximation matrix $\tilde{A}$

**Theorem 2.3.** *Algorithm ConsistApprox works correctly and for every $n \times n$ preference matrix $A$ the algorithm computes a consistent aproximation $\tilde{A}$ in $O(n^2)$ time.*

**Remark 2.4.** Algorithm ConsistApprox can easily be modified also for the case of missing values in the input preference matrix $A$. In the modification, the missing input values create no edges in the preference digraph $\mathcal{D}(A)$.

**Theorem 2.5.** *If $A$ is a preference matrix with missing values and if the order of strongly connected components in the preference digraph $\mathcal{D}(A)$ is linear, then algorithm ConsistApprox with input matrix $A$ works correctly, and the algorithm computes a consistent aproximation $\tilde{A}$ in $O(n^2)$ time.*

## Acknowledgement

The support of the Czech Science Foundation project # 14-02424S is gratefully acknowledged.