



ELSEVIER

Contents lists available at ScienceDirect

# Applied Mathematics and Computation

journal homepage: [www.elsevier.com/locate/amc](http://www.elsevier.com/locate/amc)

## Pairwise comparisons simplified

W.W. Koczkodaj<sup>a,\*</sup>, J. Szybowski<sup>b,1</sup><sup>a</sup> Computer Science, Laurentian University, Sudbury, Ontario P3E 2C6, Canada<sup>b</sup> AGH University of Science and Technology, Faculty of Applied Mathematics, al. Mickiewicza 30, 30-059 Kraków, Poland

### ARTICLE INFO

#### Keywords:

Pairwise comparisons  
Generator  
Graph  
Tree  
Algorithm

### ABSTRACT

This study examines the notion of generators of a pairwise comparisons matrix. Such approach decreases the number of pairwise comparisons from  $n \cdot (n - 1)$  to  $n - 1$ . An algorithm of reconstructing of the PC matrix from its set of generators is presented.

© 2014 Elsevier Inc. All rights reserved.

### 1. Introduction

In [17], Thurstone proposed “The Law of Comparative Judgments” for pairwise comparisons (for short, PC). However, the first use of pairwise comparisons is in [5]. Even earlier use of PC is published in [3], but in a more simplified way for voting (win or loss). The PC theory finds a lot of applications, for example, in transport. An integrated simulation, multivariate analysis and multiple decision analysis for railway system improvement and optimization is presented in [1] where data envelopment analysis is used to solve the multi-objective model to identify the best alternatives.

In many cases, we meet the incomplete PC matrices which should be completed in such a way that they become consistent. The authors of [4] deal with this problem by means of similarity and parametric compromise functions. In [15], a fitness function is defined as a scalar vector function composed of the common error measure, based on the Euclidean distance, and a minimum violation error that accounts for no violation of the rank ordering is considered to improve deriving of the weights.

In this study, we examine the possibility of reconstructing the entire  $n \times n$  PC matrix from only  $n - 1$  given entries placed in strategic locations. We call them PC-generators. Before we progress, some terminologies of pairwise comparisons must be revisited in the next section, since PC theory is still not as popular as other mathematical theories. However, the next section is definitely not for PC method experts.

### 2. Pairwise comparisons basics

We define an  $n \times n$  pairwise comparison matrix simply as a square matrix  $M = [m_{ij}]$  such that  $m_{ij} > 0$  for every  $i, j = 1, \dots, n$ . A pairwise comparison matrix  $M$  is called reciprocal if  $m_{ij} = \frac{1}{m_{ji}}$  for every  $i, j = 1, \dots, n$  (then automatically  $m_{ii} = 1$  for every  $i = 1, \dots, n$ ). Let us assume that:

\* Corresponding author.

E-mail addresses: [wkoczkodaj@cs.laurentian.ca](mailto:wkoczkodaj@cs.laurentian.ca) (W.W. Koczkodaj), [szybowski@agh.edu.pl](mailto:szybowski@agh.edu.pl) (J. Szybowski).<sup>1</sup> Research supported by the Polish Ministry of Science and Higher Education.

$$M = \begin{bmatrix} 1 & m_{12} & \cdots & m_{1n} \\ \frac{1}{m_{12}} & 1 & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m_{1n}} & \frac{1}{m_{2n}} & \cdots & 1 \end{bmatrix},$$

where  $m_{ij}$  expresses a relative quantity, intensity, or preference of entity (or stimuli)  $E_i$  over  $E_j$ . A more compact and elegant specification of PC matrix is given in [14] by Kulakowski.

A pairwise comparison matrix  $M$  is called *consistent* (or *transitive*) if:

$$m_{ij} \cdot m_{jk} = m_{ik},$$

for every  $i, j, k = 1, 2, \dots, n$ .

We will refer to it as a “consistency condition”. Consistent PC matrices correspond to the situation with the exact values  $\mu(E_1), \dots, \mu(E_n)$  for all the entities. In such case, the quotients  $m_{ij} = \mu(E_i)/\mu(E_j)$  then form a consistent PC matrix. The vector  $s = [\mu(E_1), \dots, \mu(E_n)]$  is unique up to a multiplicative constant. While every consistent matrix is reciprocal, the converse is generally false. If the consistency condition does not hold, the matrix is *inconsistent* (or *intransitive*). Axiomatization of inconsistency indicators for pairwise comparisons has been recently proposed in [13] and various inconsistency indexes are analyzed in [2].

The challenge for the pairwise comparisons method comes from the lack of consistency of the pairwise comparisons matrices, which arises in practice (while as a rule, all the pairwise comparisons matrices are reciprocal). Given an  $N \times N$  matrix  $M$ , which is not consistent, the theory attempts to provide a consistent  $n \times n$  matrix  $M'$ , which differs from matrix  $M$  “as little as possible”.

It is worth to note that the matrix:  $M = [v_i/v_j]$  is consistent for all (even random) positive values  $v_i$ . It is an important observation since it implies that a problem of approximation is really a problem of a norm selection and the distance minimization. For the Euclidean norm, the vector of geometric means (equal to the principal eigenvector for the transitive matrix) is the one which generates it. Needless to say that only optimization methods can approximate the given matrix for the assumed norm (e.g., LSM for the Euclidean distance, as recently proposed in [7]). Such type of matrices are examined in [16] as “error-free” matrices.

It is unfortunate that the singular form “comparison” is sometimes used considering that a minimum of three comparisons are needed for the method to have a practical meaning. Comparing two entities (stimuli or properties) in pairs is irreducible, since having one entity compared with itself gives trivially 1. Comparing only two entities ( $2 \times 2$  PC matrix) does not involve inconsistency. Entities and/or their properties are often called stimuli in the PC research but are rarely used in applications.

### 3. The PC-generators of pairwise comparisons matrix

For a given PC matrix  $A \in M_{n \times n}(\mathbb{R})$  consider the set  $C_n := \{a_{ij} : i < j\}$ . Note that in order to reconstruct the whole consistent matrix it is enough to know the elements of  $C_n$ , as  $a_{ii} = 1$  for each  $i \in \{1, \dots, n\}$  and  $a_{ji} = \frac{1}{a_{ij}}$  for  $i < j$ .

Let us call each such set sufficient to reconstruct the matrix  $A$  its *set of PC-generators*.

The set  $C_n$  has  $\frac{n^2-n}{2}$  elements. However, consistency is a much stronger condition. So, it is obvious that we may reduce this input set by computing the rest of elements. It is a natural question to ask which minimal subsets of  $C_n$  generate  $A$ .

**Remark 3.1.** If  $B \subset B' \subset C_n$  and  $B$  generates  $A$ , then  $B'$  does as well.

**Theorem 3.2.** There is no  $(n-2)$ -set of PC-generators of  $A$ .

**Proof.** For  $n = 3$  the statement is obvious, as in any matrix:

$$\begin{bmatrix} 1 & a & c \\ \frac{1}{a} & 1 & b \\ \frac{1}{c} & \frac{1}{b} & 1 \end{bmatrix}.$$

if we only know one of the values  $a, b$  or  $c$ , we cannot clearly calculate the other two satisfying  $c = ab$ .

To continue the induction, let us assume that the assertion holds for each matrix  $M \in M_{n \times n}(\mathbb{R})$ . Now consider the matrix:

$$A_{n+1} = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ \frac{1}{a_{12}} & 1 & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{a_{1n}} & \frac{1}{a_{2n}} & \cdots & 1 & a_{n,n+1} \\ \frac{1}{a_{1,n+1}} & \frac{1}{a_{2,n+1}} & \cdots & \frac{1}{a_{n,n+1}} & 1 \end{bmatrix}.$$

Notice that in order to calculate the elements of the last column, we need to know at least one of them. On the other hand, if we know  $p$  of them, we can calculate only  $p - 1$  of the elements  $a_{ij}$  for  $1 \leq i < j \leq n$ . Let us assume there is a  $(n - 1)$ -set  $B$  of PC-generators of  $A_{n+1}$ . We define a new set  $B' := B \cup L \setminus R$ , where  $R$  denotes the set of elements of  $B$  from the last column, and  $L$  denotes the elements from the previous columns which can be calculated from the elements of  $R$ . Now  $B'$  is a  $(n - 2)$ -set of PC-generators of the matrix  $A_n$  resulting from  $A_{n+1}$  by removing the last row and column.

$A_n \in M_{n \times n}(\mathbb{R})$ , which contradicts the inductive assumption.  $\square$

**Remark 3.3.** Given input values  $a_{i,i+1}$  (for  $i = 1, \dots, n - 1$ ) located above the main diagonal, from the consistency condition we can reconstruct the entire matrix

$$A = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ \frac{1}{a_{12}} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{a_{1n}} & \frac{1}{a_{2n}} & \cdots & 1 \end{bmatrix},$$

using the formula

$$a_{ij} = \prod_{k=i}^{j-1} a_{k,k+1}, \tag{1}$$

for  $j > i$ .

It is worth mentioning that the above  $n - 1$  values are not the only values generating the entire matrix. The following theory is provided for finding the other minimal sets of PC-generators.

**Remark 3.4.** There is a mutual relevance between each set  $B \subset C_n$  and an undirected graph  $G_B$  with  $n$  vertices:

$$a_{ij} \in B \iff \text{there is an edge } i - j \text{ in } G_B.$$

**Lemma 3.5.** If we know  $k$  edges of any subtree of  $G_B$ , then we are able to compute  $\binom{k+1}{2}$  of elements of  $C_n$ .

**Proof.** For  $k = 1$  the statement is obvious, as  $\binom{1+1}{2} = 1$ .

To proceed with induction, let us assume that the statement is true for  $k$ . Take a tree  $D$  with  $k + 2$  vertices  $x_1, \dots, x_{k+2} \in V(D)$  and  $k + 1$  edges. Remove any leaf  $x_l \in V(D)$  together with the edge  $x_m - x_l$ , joining the leaf with the tree. We get a new tree  $D'$  with  $k$  edges. From the inductive assumption, we are able to calculate  $\binom{k+1}{2}$  of elements of  $C_n$ .

Now, when we give the removed edge back, we notice that for every vertex  $x_j \in V(D) \setminus \{l\}$  there is a path  $x_j - x_{p_1} - \dots - x_{p_s} - x_l$  joining  $x_j$  with  $x_l$  and we can compute  $a_{jl} = a_{jp_1} \cdot a_{p_1 p_2} \cdot \dots \cdot a_{p_s l}$ . These are  $k + 1$  new elements and altogether we know  $\binom{k+1}{2} + k + 1 = \binom{k+2}{2}$  of elements of  $C_n$ .  $\square$

We can now formulate the necessary and sufficient condition for minimal sets of PC-generators.

**Theorem 3.6.** Let us assume  $B \subset C_n$  is a  $(n - 1)$ -set. Then  $B$  generates matrix  $A \iff G(B)$  is a tree.

**Proof.** Take a set  $B$  of PC-generators of matrix  $A$ . Let us assume  $G(B)$  is not a tree. If so, it must contain a cycle  $x_{k_1} - x_{k_2} - \dots - x_{k_s} - x_{k_1}$ . When we remove the edge  $x_{k_1} - x_{k_s}$  the relevant  $(n - 2)$ -set still generates  $A$ , as  $a_{k_1, k_s} = a_{k_1, k_2} \cdot \dots \cdot a_{k_{s-1}, k_s}$ . Thus, we get a contradiction with **Theorem 3.2**.

The reverse implication follows straight from **Lemma 3.5** for  $k = n - 1$ .  $\square$

**Corollary 3.7.** There are  $n^{n-2}$  minimal sets of PC-generators of  $A$ .

**Proof.** This is an immediate consequence of the Cayley's formula for the number of trees on  $n$  vertices.  $\square$

Although there are many combinations of  $n - 1$  values generating the entire PC matrix, the values  $m_{i,i+1}$  are the most important of all combinations since they express this sequence:

$$E_1/E_2, E_2/E_3, \dots, E_{n-2}/E_{n-1}, E_{n-1}/E_n. \tag{2}$$

Let us call the above  $n - 1$  values as *PC principal generators* (PCPGs). For matrix  $A_n$ , the PCPGs are located above the main diagonal and are as follows:

$$a_{1,2}, a_{2,3}, \dots, a_{n-1,n}.$$

Let us invent our own handicapping. Handicapping, in sports and games, is the practice of assigning advantage through scoring compensation or other advantage given to different contestants to equalize the chances of winning. The same term also applies to the various methods for computing advantages.

Entities 1 and  $n$  occurs in (2) only once. Any other entity, 2 to  $n - 1$ , occurs twice. Since the highest frequency is 2, it is fair to add 1 to  $E_1$  and  $E_n$  to compensate for only one occurrence and count the maximum number of occurrences of an entity to compensate other entities. We also define the total handicapping as the total of all compensations.

By the frequency  $f(i, B)$  of an entity  $i \in \{1, \dots, n\}$  in a set  $B \subset C_n$  we understand the cardinality of set

$$B_i := \{a_{jk} \in B : j = i \vee k = i\}.$$

Let  $O(B)$  denote the set  $\{i : B_i \neq \emptyset\}$ . We define the *total handicapping* of the set  $B \subset C_n$  as

$$h(B) := \sum_{i \in O(B)} (\max_{j \in O(B)} f(j, B) - f(i, B)).$$

For example, the total handicapping would be  $(n - 2)(n - 1)$  for all PC-generators in the first row since  $E_1$  occurs  $n - 1$  times. The rest of entities occur only once hence they need to be handicapped by  $n - 2$ .

**Remark 3.8.** The frequency of an entity  $i$  in the set  $B$  is equal to the degree of vertex  $i$  in graph  $G_B$ :

$$f(i, B) = \deg_{G_B}(i).$$

**Remark 3.9.**  $O(B)$  is the set of vertices of  $G_B$  with a positive degree.

**Remark 3.10.**  $h(B)$  counts the sum of differences between the maximal degree of a vertex in  $G_B$  and the degrees of the rest of vertices.

**Corollary 3.11.**  $h(B) = 0 \Rightarrow$  graph  $G_B$  is regular (all its vertices have the same degree).

**Theorem 3.12.** If  $n > 1$  and  $B$  is a  $(n - 1)$ -set of PC-generators of  $A$ , then

1.  $h(B) = 0 \Rightarrow n = 2$ .
2.  $h(B) \neq 1$ .
3.  $h(B) = 2 \Rightarrow G_B$  is a path connecting all vertices.

**Proof.** From Theorem 3.6, it follows that  $G(B)$  is a tree. Thus, it has  $n$  vertices and  $n - 1$  edges. Notice that to obtain the number of edges one needs to sum up the degrees of vertices and divide by two (each edge is counted twice).

If  $h(B) = 0$ , then  $G(B)$  has  $n$  vertices of the same degree  $k$ . Hence,

$$n - 1 = \frac{nk}{2} \Rightarrow 2n - 2 = nk \Rightarrow n(2 - k) = 2 \Rightarrow n = 2.$$

If  $h(B) = 1$ , then  $G(B)$  has  $n - 1$  vertices of the same degree  $k$  and one of degree  $k - 1$ . Hence,

$$n - 1 = \frac{(n - 1)k}{2} + \frac{k - 1}{2} = \frac{nk - 1}{2} \Rightarrow 2n - 2 = nk - 1 \Rightarrow (2 - k)n = 1 \Rightarrow n = 1,$$

which is a contradiction.

If  $h(B) = 2$ , then there are two cases. The first one with  $n - 1$  vertices of degree  $k$  and one vertex of degree  $k - 2$ . Hence,

$$n - 1 = \frac{(n - 1)k}{2} + \frac{k - 2}{2} = \frac{nk - 2}{2} \Rightarrow 2n - 2 = nk - 2 \Rightarrow k = 2,$$

and it follows that there is one vertex of degree 0, so graph  $G_B$  is disconnected, which is a contradiction.

For a case with  $n - 2$  vertices of degree  $k$  and two vertices of degree  $k - 1$ , we have:

$$n - 1 = \frac{(n - 2)k}{2} + \frac{2}{k - 1} = \frac{nk - 2}{2} \Rightarrow 2n - 2 = nk - 2 \Rightarrow k = 2,$$

and it follows that  $G_B$  is a path connecting all vertices.  $\square$

**Corollary 3.13.** For  $n \geq 2$  there are  $\frac{n!}{2} (n - 1)$ -sets of PC-generators of  $A$  with the minimal total handicapping.

**Proof.** For  $n = 2$ , the statement is obvious. For  $n \geq 3$  from Theorem 3.12, we know that it suffices to count all the paths connecting all vertices. Each such path is of the form  $p(1) - p(2) - \dots - p(n)$ , where  $p$  is a permutation of the set  $\{1, \dots, n\}$ . However, from  $p(1) - p(2) - \dots - p(n)$  and  $p(n) - p(n - 1) - \dots - p(1)$ , we obtain the same tree hence the number of permutations must be divided by two.  $\square$

**Example 3.14.** Consider  $n = 4$  and the matrix

$$A = \begin{bmatrix} 1 & a & b & c \\ \frac{1}{a} & 1 & d & e \\ \frac{1}{b} & \frac{1}{d} & 1 & f \\ \frac{1}{c} & \frac{1}{e} & \frac{1}{f} & 1 \end{bmatrix}.$$

The set  $C_4 = \{a, b, c, d, e, f\}$  consists of  $\frac{4^2 - 4}{2} = 6$  elements, so it has  $\binom{6}{3} = 20$  3-subsets listed in Table 1. From Remark 3.4, we know that they are related to some graphs. They are listed in Fig. 1 and their nodes are labeled in Fig. 2.

The last four subsets are triads, so they are not PC-generators of  $A$ . The graphs related to them are cycles.

According to Corollary 3.7, there are  $4^{4-2} = 16$  minimal sets of PC-generators of  $A$  and they are the first 16. The graphs related to them are obviously trees. From Corollary 3.13, we conclude that we have  $\frac{4!}{2} = 12$  sets of PC-generators of  $A$  minimizing the total handicapping and they are the first 12. The related graphs are paths.

It is worth noticing that the first subset is the set of PGs.

The strategic importance of PGs, based on the total handicapping, places them on the diagonal above the main diagonal. It is important to notice that PGs are always there and it is up to us to decide whether or not we use them for reconstructing the entire consistent PC matrix (with the possible increased error) or enter the remaining entries into PC matrix with the possibility of inconsistency in triads. As pointed out in [11,6,8,9,10,12], nothing is carved in stone and even reciprocity can be relaxed for the blind wine and other types of tasting. Reciprocity may also not be guaranteed when data are collected over the Internet from different sources or individuals. For PGs sufficiently accurate, the PC matrix reconstruction is a vital possibility.

#### 4. An algorithm for the reconstruction of a PC matrix from a minimal set of PC-generators

Assuming that we know the  $(n - 1)$ -set  $B \subset C_n \subset A$ , we can reconstruct the entire matrix  $A$  using an algorithm described below.

For given PGs  $a_{11}, \dots, a_{nn}$ , we can compute each  $a_{ij}$  for  $i < j$  from Remark 3.3. On the other hand, if we know a  $(n - 1)$ -set  $B$  of different values of matrix  $A$  we can apply the log function to the formula (1) and we substitute  $x_k := \log a_{k,k+1}$  then we get the system of linear equations

$$\log a_{ij} = \sum_{k=i}^{j-1} x_k, \quad a_{ij} \in B.$$

If  $B$  generates  $A$ , then this system has a unique solution and we may calculate  $a_{k,k+1} = 10^{x_k}$ .

This leads us to the following algorithm:

**INPUT:**

- $M \in \mathcal{M}(n - 1, 3)$  representing the set  $B$  of PC-generators of  $A$ .
- $(M[i, 1], M[i, 2])$  correspond to the coordinates of the  $i$ th element of  $B$ .
- $M[i, 3]$  corresponds to the value of the  $i$ th element of  $B$ .

**OUTPUT:**

The consistent PC matrix  $A \in \mathcal{M}(n, n)$  reconstructed from  $A$ .

**Table 1**  
3-subsets for  $n = 4$ .

$\{a, d, f\}$	$\{a, c, d\}$	$\{a, c, f\}$	$\{c, d, f\}$
$\{a, e, f\}$	$\{a, b, f\}$	$\{b, c, d\}$	$\{c, d, e\}$
$\{a, b, e\}$	$\{b, c, e\}$	$\{b, e, f\}$	$\{b, d, e\}$
$\{a, b, c\}$	$\{a, d, e\}$	$\{b, d, f\}$	$\{c, e, f\}$
$\{a, c, e\}$	$\{a, b, d\}$	$\{b, c, f\}$	$\{d, e, f\}$

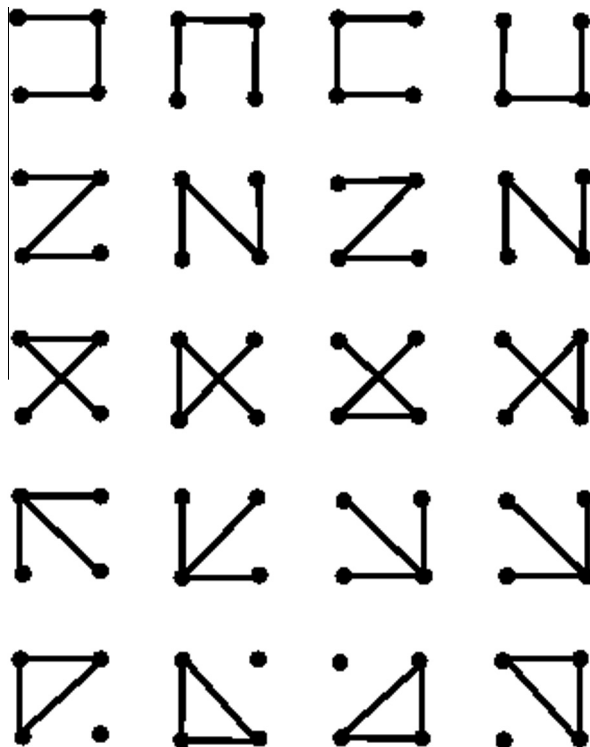


Fig. 1. All possible combinations of 20 three PC-generators.

**ALGORITHM:**

1. Construct graph  $G_B$ :  
 $G_B := \emptyset$ ;  
 for  $i := 1$  to  $n - 1$   $G_B := G_B \cup \{M[i, 1] - M[i, 2]\}$ ;
2. Check if  $G_B$  is a tree (using DFS algorithm): if not then write ('B does not generate A') and exit;
3. Solve a linear system

$$\sum_{k=M[i,1]}^{M[i,2]-1} x_k = \log(M[i, 3]), \quad i = 1, \dots, n - 1.$$

4. Calculate elements of A:  
 for  $k := 1$  to  $n$   $A[k, k] := 1$ ;  
 for  $k := 1$  to  $n - 1$   $A[k, k + 1] := 10^{x_k}$ ;  
 for  $k := 1$  to  $n - 1$   
     for  $l := k + 2$  to  $n$   $A[k, l] := \prod_{m=k}^{l-1} A[m, m + 1]$   
 for  $k := 1$  to  $n - 1$   
     for  $l := k + 1$  to  $n$   $A[l, k] := \frac{1}{A[k, l]}$

**Example 4.1.** Let us consider the PC matrix A, from the Example 3.14, and assume that we know the values from the set  $B = \{a, b, e\}$ . Thus, the input matrix is:

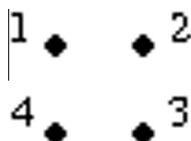


Fig. 2. Nodes for all PC-generators in Fig. 1.

$$M = \begin{bmatrix} 1 & 2 & a \\ 1 & 3 & b \\ 2 & 4 & e \end{bmatrix}.$$

Apply the algorithm:

1. We obtain the graph  $G_B$  with 4 vertices

1, 2, 3, 4,

and 3 edges

1 – 2, 1 – 3, 2 – 4.

2. We assure that  $G_B$  is a tree and continue.

3. We solve the linear system

$$\begin{cases} x_1 = \log a \\ x_1 + x_2 = \log b, \\ x_2 + x_3 = \log e \end{cases}$$

and we get

$$\begin{cases} x_1 = \log a \\ x_2 = \log b - \log a = \log \frac{b}{a} \\ x_3 = \log e - \log b + \log a = \log \frac{ae}{b} \end{cases}.$$

4. We calculate the missing elements of the output PC matrix  $A$ :

$$A = \begin{bmatrix} 1 & a & b & ae \\ \frac{1}{a} & 1 & \frac{b}{a} & e \\ \frac{1}{b} & \frac{a}{b} & 1 & \frac{ae}{b} \\ \frac{1}{ae} & \frac{1}{e} & \frac{b}{ae} & 1 \end{bmatrix}.$$

**Remark 4.2.** The complexity of the algorithm is  $O(n^3)$ .

**Proof.** Steps 1 and 2 take  $O(n)$  operations. The complexity of step 3 is  $O(n^2)$  and of step 4 is  $O(n^3)$ .  $\square$

Notice that the assumption that the number  $p$  of input entries is equal to  $n - 1$  may not be satisfied. We may have less or more data and still use the slightly modified algorithm. We only change steps **1** and **2**:

1. Construct graph  $G_B$ :

$G_B := \emptyset$ ;

for  $i := 1$  to  $p$   $G_B := G_B \cup \{M[i, 1] - M[i, 2]\}$ ;

2. Replace  $G_B$  by its spanning tree (which is at most a  $(n - 1)$ -set) received from DFS algorithm.

if  $G_B$  has less than  $n - 1$  edges then write (' $B$  does not generate  $A$ ') and exit.

## 5. Conclusions

In this study, we have demonstrated that we are able to use less than all  $n \cdot (n - 1)/2$  pairwise comparisons to reconstruct a consistent matrix. In fact, the minimum of  $n - 1$  PC-generators is sufficient. This may be very useful, provided that they represent more accurate values than the rest of the PC matrix. Another possible application of the suggested algorithm is a case of the missing data in the PC matrix. The entire PC can be reconstructed from the PCPGs located above the main diagonal. Together with the preliminary sorting of entities, it is a quick and simple method of getting weights. The problem is that the accumulated errors grow fast when we "walk away" from the main diagonal towards the upper right corner. In fact, for  $n = 7$  and 20% error in the PCPGs, nearly 200% is accumulated in the upper right corner. The proposed algorithm for reconstructing the entire matrix from the principal PC-generators is easy to implement, even in Gnumeric (or in MS Excel). In addition, the reconstructed values (accurate or not) are consistent. In this study, we only consider the multiplicative variant of PC, which is based on "how many times?", while the additive version of pairwise comparisons ("by how much?") was recently analyzed in [18]. The additive PC method utilizes a different type of inconsistency (not addressed here). Certainly more research is needed.

An interesting problem is how to reconstruct the matrix from a set  $B$  containing more than  $n - 1$  elements. In such a case, the corresponding graph  $G_B$  must have at least one cycle, so we may process the redundant information. There is a unique extension of  $B$  to a consistent matrix  $A$  if the condition:

$$a_{i_1 i_2} \cdot a_{i_2 i_3} \cdot \dots \cdot a_{i_{s-1} i_s} \cdot a_{i_s i_1} = 1$$

is satisfied for each path (including cycles) of  $G_B$ . Thus, it is sufficient to take any spanning tree  $T$  of  $G_B$  and reconstruct  $A$  from  $T$ , which is a  $(n - 1)$ -set. Unfortunately, there will be multiple solutions corresponding to various  $(n - 1)$ -subsets of  $B$  if the consistency condition is not satisfied.

The problem of selecting the resulting PC matrix is an interesting challenge for a separate publication. An ad hoc solution would be the choice of the  $(n - 1)$ -subset of  $B$ , minimizing the total handicapping. Another approach is to use the *HRE* algorithm, introduced in [14]. In many cases, the application domain (e.g., medical data analysis) may have its own requirements to follow.

## Acknowledgments

The second author has been supported in part by the Euro Research grant “Human Capital”. Authors would like to thank Amanda Dion-Groleau (Laurentian University, Psychology) and Grant O. Duncan (a part-time graduate student at Laurentian University and Team Lead, Business Intelligence and Software Integration, Health Sciences North, Sudbury, Ontario) for their help with proofreading this text.

## References

- [1] A. Azadeh, S.F. Ghaderi, H. Izadbakhsh, *Appl. Math. Comput.* 195 (2) (2008) 775–785.
- [2] M. Brunelli, L. Canal, M. Fedrizzi, Inconsistency indices for pairwise comparison matrices a numerical study, *Ann. Oper. Res.* 211 (1) (2013) 493–509.
- [3] N. de Condorcet, *Essay on the Application of Analysis to the Probability of Majority Decisions*, l’Imprimerie Royale, Paris, 1785.
- [4] E. Dopazo, M. Ruiz-Tagle, A parametric GP model dealing with incomplete information for group decision-making, *Appl. Math. Comput.* 218 (2) (2011) 514–519.
- [5] G.T. Fechner, *Elements of Psychophysics*, vol. 1, Holt, Rinehart & Winston, New York, 1965, translation by H.E. Adler of *Elemente der Psychophysik*, Breitkopf und Härtel, Leipzig, 1860.
- [6] J. Fulop, W.W. Koczkodaj, S.J. Szarek, A different perspective on a scale for pairwise comparisons, *Trans. Comput. Collective Intell. Lect. Notes Comput. Sci.* 6220 (2010) 71–84.
- [7] A.Z. Grzybowski, Note on a new optimization based approach for estimating priority weights and related consistency index, *Expert Syst. Appl.* 39 (14) (2012) 11699–11708.
- [8] M. Herman, W.W. Koczkodaj, Monte Carlo study of pairwise comparisons, *Inf. Process. Lett.* 57 (1) (1996) 25–29.
- [9] W.W. Koczkodaj, A new definition of consistency of pairwise comparisons, *Math. Comput. Model.* 18 (7) (1993) 79–84.
- [10] W.W. Koczkodaj, Testing the accuracy enhancement of pairwise comparisons by a Monte Carlo experiment, *J. Stat. Planning Inference* 69 (1) (1998) 21–32.
- [11] W.W. Koczkodaj, M. Orłowski, Computing a consistent approximation to a generalized pairwise comparisons matrix, *Comput. Math. Appl.* 37 (1999) 79–85.
- [12] W.W. Koczkodaj, S.J. Szarek, On distance-based inconsistency reduction algorithms for pairwise comparisons, *Logic J. IGPL* 18 (6) (2010) 859–869.
- [13] W.W. Koczkodaj, R. Szwarc, On Axiomatization of Inconsistency Indicators for Pairwise Comparisons, *Fundam. Inform.* 132 (4) (2014) 485–500.
- [14] K. Kulakowski, A heuristic rating estimation algorithm for the pairwise comparisons method, *Eur. J. Oper. Res.* (online) 2013.
- [15] B. Srdjevic, Z. Srdjevic, Bi-criteria evolution strategy in estimating weights from the AHP ratio-scale matrices, *Appl. Math. Comput.* 218 (4) (2011) 1254–1266.
- [16] J. Temesi, Pairwise comparison matrices and the error-free property of the decision maker, *Cent. Eur. J. Oper. Res.* 19 (2) (2011) 239–249.
- [17] L.L. Thurstone, A law of comparative judgments, *Psychol. Rev.* 34 (1927) 273–286.
- [18] K.K.F. Yuen, Pairwise opposite matrix and its cognitive prioritization operators: comparisons with pairwise reciprocal matrix and analytic prioritization operators, *J. Oper. Res. Soc.* 63 (3) (2012) 322–338.