European Journal of Operational Research 257 (2017) 197–208

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Decision Support

The mathematical equivalence of the “spanning tree” and row geometric mean preference vectors and its implications for preference analysis

Michele Lundy[a,b], Sajid Siraj[c,d,*], Salvatore Greco[b,e]

[a] Dublin City University, Dublin, Republic of Ireland
[b] Portsmouth Business School, University of Portsmouth, Portsmouth, UK
[c] Centre for Decision Research, Leeds University Business School, Leeds, UK
[d] COMSATS Institute of Information Technology, Wah Cantonment, Pakistan
[e] Department of Economics and Business, University of Catania, Italy

ARTICLE INFO

Article history:
Received 1 May 2015
Accepted 19 July 2016
Available online 27 July 2016

Keywords:
Decision analysis
Pairwise comparisons
Multiple criteria analysis
Graph theory
Spanning trees

ABSTRACT

Pairwise comparison is a widely used approach to elicit comparative judgements from a decision maker (DM), and there are a number of methods that can be used to then subsequently derive a consistent preference vector from the DM’s judgements. While the most widely used method is the eigenvector method, the row geometric mean approach has gained popularity due to its mathematical properties and its ease of implementation. In this paper, we discuss a spanning tree method and prove the mathematical equivalence of its preference vector to that of the row geometric mean approach. This is an important finding due to the fact that it identifies an approach for generating a preference vector which has the mathematical properties of the row geometric mean preference vector, and yet, in its entirety, the spanning tree method has more to offer than the row geometric mean method, in that, it is inherently applicable to incomplete sets of pairwise comparison judgements, and also facilitates the use of statistical and visual techniques to gain insights into inconsistency in the DM’s judgements.

1. Introduction

Pairwise comparison (PC) is a widely used approach to elicit comparative judgements from a decision maker (DM). In the PC method, the DM is asked a series of questions to compare the available options in pairs, and eventually, a prioritization method is applied to these judgements in order to estimate the DM’s preferences in the form of a preference vector. The preference vector is a vector of weights representing the relative strength of preferences for available options. However, since the judgements acquired from the DM often contain inconsistency, the process of estimating a preference vector is not necessarily straightforward. Inconsistency occurs when the direct comparative value of a pair of options does not match the indirect comparative value derived from an intermediate third option. For example, if option A is declared twice as preferred as option B and option B is declared three times as preferred as option C, then the indirect comparative value suggests that option A be preferred six times more than option C and yet the DM may directly declare option A to be say five times as preferred as option C, which is obviously inconsistent with the other two comparative judgements. That is the direct comparative value of Option A and Option C (i.e. 5) does not match the indirect comparative value of Option A and Option C derived from an intermediate third option B (i.e. 6). Of course, the number of comparisons increases with the number of options which, in turn, increases the possibility of having at least some and possibly a high number of inconsistent comparisons. Therefore, any prioritization method must be able to estimate the preference vector from an inconsistent set of comparisons.

Historically, the principal right eigenvector (REV) prioritization method (Saaty, 1977) has been widely used for estimating the preference vector for both consistent and (acceptably) inconsistent PC judgements where, in the REV method, the PC judgements are used to construct a PC matrix, the principal eigenvector of which is taken as the preference vector. The inconsistency is measured in terms of the Consistency Ratio (CR) which is an Eigenvalue based measure with the PC matrix only considered acceptable if the CR value remains below a certain limit (usually $CR < 0.1$).

[*] Corresponding author at: Centre for Decision Research, Leeds University Business School, Leeds, UK. Tel.: +44 11334 39652.

E-mail addresses: michele.lundy@port.ac.uk (M. Lundy), sajidsiraj@gmail.com, s.siraj@leeds.ac (S. Siraj), salgreco@unict.it (S. Greco).

http://dx.doi.org/10.1016/j.ejor.2016.07.042
0377-2217/© 2016 Elsevier B.V. All rights reserved.

Johnson (1979) discovered that, for the same problem, the use of left eigenvectors may produce a different solution to that of the right eigenvector approach, yet considered the use of left eigenvectors to be as equally justified as the use of right eigenvectors. Therefore, the REV method has been criticized due to this left-right eigenvector asymmetry, the use of arbitrary thresholds for inconsistency acceptability, as well as a few other further issues (Bana e Costa & Vansnick, 2008; Barzilai, 1997; Barzilai, Cook, & Golany, 1987). Due to these shortcomings, several other prioritization methods for preference vector estimation have been proposed in the literature which also begin by constructing a PC matrix from the PC judgements. For example, the logarithmic least squares (LLS) method, proposed in Crawford and Williams (1985), assumes that the most preferred approach for prioritization is to find the vector that minimizes the sum of the logarithmic residuals from a given set of judgements. Considering the multiplicative properties of PC, Crawford and Williams (1985) showed that the LLS method always generates a unique solution, and in the case of a complete set of PC judgements, the LLS solution is identical to the solution calculated using the row geometric mean (RGM) of the constructed PC matrix. In addition to these approaches, there exists a number of other optimization-based methods like direct least squares (DLS) (Chu, Kalaba, & Spingarn, 1979), logarithmic least absolute value (LLAV) (Cook & Kress, 1988), and fuzzy preference programming (Mikhailov, 2000). Choo and Wedley (2004) analysed and numerically compared a variety of these prioritization methods and concluded that there is no single best method that outperforms the others in every situation.

Although REV is the most commonly used method, the RGM approach has gained popularity due to its mathematical properties, and while shown to be equivalent to the LLS approach (Crawford & Williams, 1985), RGM has additional benefits due to its ease of implementation (Crawford, 1987; Williams & Crawford, 1980). Indeed (Williams & Crawford, 1980) proposed using the RGM method rather than the REV method due to its ease of computation, and also demonstrated its advantages arising from common statistical and mathematical properties. Since the objective of the prioritization method is to obtain a single preference vector from an inconsistent PC matrix, most methods therefore justifiably focus on this aspect, and therefore assess inconsistency only by measuring it for the purpose of accepting or rejecting the provided PC judgements as suitable rather than analysing inconsistency. That is, while focusing on this "single solution" aspect, an in-depth analysis of the inconsistency is neglected.

We contend that a prioritization method must have the capabilities to focus on both aspects of the problem, i.e. production of a single "good quality" preference vector and also facilitation of an in-depth inconsistency analysis. The latter aspect is illustrated in Section 4.1 by establishing an underlying universe of potential preference vectors and then examining the degree of homogeneity within them. In this way we can start to unravel any inconsistency in the decision maker's judgements by translating inconsistency into a number of different possible mindsets. This is important particularly of course when inconsistency is high and so where the DM may need significant help to resolve his/her inconsistency, but also sometimes even when CR is low, as situations can arise where even though the CR value might otherwise be regarded as acceptably low, it is clear that using this acceptability criterion may be quite inappropriate - see illustration in Section 4.1.

Also, Harker (1987b) investigated incomplete sets of judgements where the DMs are allowed to respond with "do not know" or "not sure" to some judgements. This is an important issue to investigate as the probability of acquiring an incomplete set of PC judgements increases with an increase in the total number of items for comparison (Fedrizzi & Giove, 2007, 2013; Schubert, 2014). Both the REV and the RGM methods are inappropriate in such cases due to the fact that the PC matrix cannot be constructed without estimating/imputing the missing judgements (see Section 4.2 for details).

Indeed, several criteria have been suggested to compare prioritization methods in the literature. For example, minimal deviation from the DM's judgements (Kou & Lin, 2014; Lin, 2007; Siraj, Mikhailov, & Keane, 2012b), computational complexity, ability to handle incomplete sets of judgements (Ergu, Kou, Peng, Shi, & Shi, 2011; Harker, 1987a; Srdjevic, Srdjevic, & Blagojevic, 2014), adhering to geometric properties (Aguaron & Moreno-Jimenez, 2003; Barzilai, 1997), and ability to measure inconsistency (Brunelli, Canal, & Fedrizzi, 2013; Brunelli & Fedrizzi, 2015; Tomashevskii, 2015). While there is no consensus with regards to which of these "conventional" performance measures should be used for comparative assessment, we contend that a prioritization method should meet as many of these criteria as possible, and must also have the ability to facilitate the analysis of inconsistency.

In this context, a graph-theoretic approach was recently formulated to calculate a preference vector by taking the average of all possible preference vectors calculated through enumeration of all possible spanning trees (EAST) (Tsyganok, 2010; see also Siraj, Mikhailov, & Keane, 2012a). The proposed method was shown to have a number of desirable properties including, for example, producing a solution with minimal deviation from the PC judgements and measuring the level of inconsistency in these judgements. However, since the original method used the arithmetic mean to calculate the average, it failed to satisfy the criterion of adhering to geometric properties. We have therefore investigated the use of the geometric mean of all "spanning tree" preference vectors (GMAST).

In this paper, we report on the quality of the GMAST method's preference vector and its adherence to the conventional performance criteria, and provide some initial insights into its capability to facilitate the analysis of inconsistency. We therefore focus on the GMAST preference vector and prove its mathematical equivalence to that of the RGM method. This is an important finding due to the fact that it establishes the quality of the GMAST preference vector by proving that it has the mathematical properties of the RGM preference vector and yet, the GMAST method in its entirety has additional benefits. That is, unlike RGM, the GMAST method is inherently applicable to incomplete PC matrices (see Section 4.2), and also facilitates in-depth inconsistency analysis (see Sections 4.1 and 6). Indeed, with respect to all of the performance criteria, the GMAST method in its entirety outperforms all the other existing prioritization methods.

2. Problem formulation

Assume that we are interested in determining a preference vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ where $\frac{w_i}{w_j}$ represents the DM's relative preference for element i compared to element j. Because we are only interested in the ratio $\frac{w_i}{w_j}$, $\mathbf{w}$ is not unique and there is a class of equivalent vectors satisfying our requirement where any member of the class only differs from another member by a multiplicative scalar.

Assuming that $A = [a_{ij}]$ is the DM's PC matrix (i.e. a_{ij}= the acquired DM's judgement for element i compared to element j), then the objective of a prioritization method is to derive a $\mathbf{w}$ from A.

Since $a_{ii} = 1$ for all $i = 1, 2, \ldots, n,$ we have

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} & \ldots & a_{1n} \\ a_{21} & 1 & a_{23} & \ldots & a_{2n} \\ a_{31} & a_{32} & 1 & \ldots & \ldots \\ \ldots & \ldots & \ldots & 1 & \ldots \\ a_{n1} & a_{n2} & \ldots & \ldots & 1 \end{bmatrix}$$

and $\mathbf{w} = f(A)$ for some formulation f, where f is essentially the prioritization method.

There are many ways of deriving a $\mathbf{w}$. For example, we could choose the k^{th} column $\mathbf{a_k} = (a_{1k}, a_{2k}, ..., a_{nk})^T$ of A and use this as a preference vector $\mathbf{w_k}$. The problem is that there is no reason why any column of A should be more appropriate than any other.

Fortunately, if A is consistent (i.e. $a_{ij} = a_{ik}a_{kj} \forall i, j, k$) then every $\mathbf{w_k}$ derived from a column $\mathbf{a_k}$ is equivalent (i.e. only differs from any other by a scalar) and so all $\mathbf{w_k}$ are equivalent to a single preference vector $\mathbf{w}$ and therefore any column can be chosen to derive $\mathbf{w}$. However, if A is inconsistent (i.e. $\exists\, i, j$ for which $a_{ij} \neq a_{ik}a_{kj}$ for some k) then the $\mathbf{w_k}$ represented by the columns $\mathbf{a_k}$ of A are not all equivalent and have to be amalgamated in some way to form a preference vector estimate $\hat{\mathbf{w}}$.

For example, using the RGM approach, we obtain the following:

$$\hat{\mathbf{w}} = \left(\hat{w}_i\right) \text{ where } \hat{w}_i = \left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}} \tag{1}$$

or using the GMAST approach, we obtain the following:

$$\hat{\mathbf{w}} = \left(\hat{w}_i\right) \quad \text{where } \hat{w}_i = \left(\prod_{\tau=1}^{\eta} \hat{w}_{i(\tau)}\right)^{\frac{1}{\eta}} \tag{2}$$

where $\hat{w}_{i(\tau)}$ are the preference weights in the preference vector $\hat{\mathbf{w}}_\tau = (\hat{w}_{i(\tau)})$ derived from spanning tree τ and where $\eta = n^{n-2}$.

In fact, there are many approaches that can be used to generate $\hat{\mathbf{w}}$ and comparing their properties is a subject of much debate. As a result of this need for comparison and the fact that any $\hat{\mathbf{w}}$ can be represented in several equivalent forms, it is usual to normalize $\hat{\mathbf{w}}$ in some way. The two most popular forms being ideal-mode and distributed-mode where if $\hat{\mathbf{u}} = (\hat{u}_i)$ represents the normalized version then, in ideal-mode:

$$\hat{\mathbf{u}} = \frac{\hat{\mathbf{w}}}{\hat{w}_1} = \left(1, \frac{\hat{w}_2}{\hat{w}_1}, \frac{\hat{w}_3}{\hat{w}_1}, ..., \frac{\hat{w}_n}{\hat{w}_1}\right) \tag{3}$$

i.e. $\hat{\mathbf{u}}$ is $\hat{\mathbf{w}}$ normalized to have w_1 as a reference, and in distributed-mode:

$$\hat{\mathbf{u}} = \frac{\hat{\mathbf{w}}}{\sum_i \hat{w}_i} = \left(\frac{\hat{w}_1}{\sum_i \hat{w}_i}, \frac{\hat{w}_2}{\sum_i \hat{w}_i}, ..., \frac{\hat{w}_n}{\sum_i \hat{w}_i}\right)$$

i.e. $\hat{\mathbf{u}}$ is $\hat{\mathbf{w}}$ normalized to have the sum of all weights equal to 1 (i.e. $\sum \hat{u}_i = 1$).

For example, in the case of RGM, since from (1) with $i = 1$, we have $\hat{w}_1 = (\prod_{j=1}^{n} a_{1j})^{\frac{1}{n}}$. Then, using (3) for normalization, we have ideal-mode RGM is $\hat{\mathbf{u}} = (\hat{u}_i)$ where:

$$\hat{u}_i = \frac{\hat{w}_i}{\hat{w}_1} = \frac{\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}}}{\left(\prod_{j=1}^{n} a_{1j}\right)^{\frac{1}{n}}} = \left(\prod_{j=1}^{n} \frac{a_{ij}}{a_{1j}}\right)^{\frac{1}{n}}$$

which by reciprocity gives:

$$\hat{u}_i = \left(\prod_{j=1}^{n} a_{ij}a_{j1}\right)^{\frac{1}{n}} \tag{4}$$

In order to prove the equivalence of the RGM preference vector and GMAST's preference vector, we initially focus on the ideal modes equivalence before generalizing the equivalence to any mode. We see later that this generalization is straight forward since for any ratio-based preference vector $\hat{u}_i/\hat{u}_j = \hat{w}_i/\hat{w}_j$ and so equivalence is unaffected by the mode of representation.

However, before proving the RGM and GMAST preference vectors equivalence, we discuss some fundamentals of the "spanning tree" approach.

2.1. Fundamentals of the "spanning tree" approach for pairwise comparisons

It is important to note that, because of reciprocity, the PC matrix $A = [a_{ij}]$ contains only $\frac{n(n-1)}{2}$ information bearing values a_{ij} (which without loss of generality can be taken as the $\frac{n(n-1)}{2}$ a_{ij} in the lower triangle of A).

And that these $\frac{n(n-1)}{2}$ a_{ij} can be represented as a complete graph with n nodes (one node per element and one edge per a_{ij}).

And there are $\eta = n^{n-2}$ spanning trees of this graph (Cayley's theorem) where each spanning tree τ consists of a subset $\mathcal{E}_\tau$ of $(n-1)$ of the graph's $\frac{n(n-1)}{2}$ edges where the spanning tree τ connects node i to node j either

- directly by the edge $[i \rightarrow j]$ (if $[i \rightarrow j] \in \mathcal{E}_\tau$), or
- indirectly by a path of edges (say $[i \rightarrow k_1 \rightarrow k_2 \rightarrow \cdots \rightarrow k_s \rightarrow j]$) where $\{[i \rightarrow k_1], [k_1 \rightarrow k_2], ..., [k_s \rightarrow j]\} \subseteq \mathcal{E}_\tau$ (if $[i \rightarrow j] \notin \mathcal{E}_\tau$).

There are therefore $(n-1)$ direct edges $[i \rightarrow j]$, and therefore there must be $\frac{n(n-1)}{2} - (n-1) = \frac{(n-1)(n-2)}{2}$ indirect paths $[i \rightarrow k_1 \rightarrow k_2 \rightarrow \ldots \rightarrow k_s \rightarrow j]$.

Therefore mapping each $[i \rightarrow j] \in \mathcal{E}_\tau$ to the corresponding a_{ij} and letting $\mathcal{A}_\tau = \left\{a_{ij} : [i \rightarrow j] \in \mathcal{E}_\tau\right\}$.

Then $[i \rightarrow j] \in \mathcal{E}_\tau \iff a_{ij} \in \mathcal{A}_\tau$ and $|\mathcal{E}_\tau| = |\mathcal{A}_\tau| = (n-1)$

That is $\mathcal{A}_\tau$ defines a sufficient subset of $(n-1)$ of the $\frac{n(n-1)}{2}$ a_{ij} in the lower triangle of A, and so, analogously to the spanning trees ability to connect any node i to any node j (directly or indirectly), $\mathcal{A}_\tau$ can be used to construct the lower triangle of an artificial PC matrix $\hat{A}_\tau$ of $\frac{n(n-1)}{2}$ $\hat{a}_{ij(\tau)}$ in which $\hat{a}_{ij(\tau)}$ is either

- set directly as a_{ij} (if $[i \rightarrow j] \in \mathcal{E}_\tau$ or equivalently, if $a_{ij} \in \mathcal{A}_\tau$), or
- derived indirectly as a transitive product of some a_{ij} (say $a_{ik_1}a_{k_1k_2}\ldots a_{k_sj}$) where $\left\{a_{ik_1}, a_{k_1k_2}, \ldots, a_{k_sj}\right\} \subseteq \mathcal{A}_\tau$ (if $[i \rightarrow j] \notin \mathcal{E}_\tau$ or equivalently, if $a_{ij} \notin \mathcal{A}_\tau$)

There are therefore $(n-1)$ direct $\hat{a}_{ij(\tau)}$ in the lower triangle of $\hat{A}_\tau$, and therefore there must be $\frac{(n-1)(n-2)}{2}$ indirect $\hat{a}_{ij(\tau)}$ in the lower triangle of $\hat{A}_\tau$ (corresponding respectively to the $(n-1)$ direct edges and the $\frac{(n-1)(n-2)}{2}$ indirect paths within the spanning tree τ).

And so setting $\hat{a}_{ii(\tau)} = 1 \forall i = 1, 2, ..., n$ and using reciprocity to derive the upper triangle of $\hat{A}_\tau$, we can construct a necessarily consistent (artificial) PC matrix $\hat{A}_\tau$ with

$$\hat{A}_\tau = \begin{bmatrix} 1 & \hat{a}_{12} & \hat{a}_{13} & \ldots & \hat{a}_{1n} \\ \hat{a}_{21} & 1 & \hat{a}_{23} & \ldots & \hat{a}_{2n} \\ \hat{a}_{31} & \hat{a}_{32} & 1 & \ldots & \ldots \\ \ldots & \ldots & \ldots & 1 & \ldots \\ \hat{a}_{n1} & \hat{a}_{n2} & \ldots & \ldots & 1 \end{bmatrix}$$

3. The equivalence of the RGM and GMAST preference vectors

Before finalizing the RGM/GMAST equivalence proof, we establish a number of supporting propositions for GMAST below.

Proposition 1. *The GMAST preference vector $\hat{\mathbf{w}}$ can be represented in ideal mode as $\hat{\mathbf{u}} = \left(\hat{u}_i\right)$ where:*

$$\hat{u}_i = \left(\prod_{\tau=1}^{\eta} \hat{a}_{i1(\tau)}\right)^{\frac{1}{\eta}}$$

and each $\hat{a}_{i1(\tau)}$ is either

- *set directly as a_{i1} (if $a_{i1} \in \mathcal{A}_\tau$), or*

- *derived indirectly as a transitive product of a subset of a_{ij}, say $a_{ik_1}a_{k_1k_2}\dots a_{k_s1}$, where $\left\{a_{ik_1,}a_{k_1k_2},\dots,a_{k_s1}\right\}\subseteq\mathcal{A}_\tau$(if $a_{i1}\notin\mathcal{A}_\tau$).*

Comment. Notice that the product term in the Proposition 1 equation contains reference to the first column of $\hat{A}_\tau$ (i.e. $\hat{a}_{i1(\tau)}$) only. This is because each $\hat{A}_\tau$ is consistent and so any column of $\hat{A}_\tau$ (including the first column) represents the preference vector $\hat{\mathbf{u}}_\tau$ of $\hat{A}_\tau$. Since the first column is naturally in ideal mode, it makes sense to choose this column. The direct/indirect categorization of $\hat{a}_{i1(\tau)}$ is using the result already established in Section 2.1. See proof of Proposition 1 in Appendix B.

Proposition 2. *Each $\hat{a}_{ij(\tau)}$ is defined directly as a_{ij} in $2n^{n-3}$ of the n^{n-2} $\hat{A}_\tau$ (and therefore indirectly in the remaining $(n-2)n^{n-3}$ $\hat{A}_\tau$).*

Comment. The proof of Proposition 2 follows from the fact that each $\hat{a}_{ij(\tau)}$ is set directly as a_{ij} if and only if the corresponding a_{ij} is chosen to form the underlying sufficient subset $\mathcal{A}_\tau$ and the fact that any given a_{ij} is chosen to form the underlying sufficient subset $\mathcal{A}_\tau$ in exactly $2n^{n-3}$ of the subsets $\mathcal{A}_\tau$. See proof of Proposition 2 in Appendix B.

Corollary to Propositions 1 and 2

$$\hat{u}_i=(a_{i1})^{\frac{2}{n}}\left[\left(\prod_{\tau=1}^{(n-2)n^{n-3}}\hat{a}_{i1(\tau)}\right)\right]^{\frac{1}{\eta}}$$

Comment. The proof of the Corollary to Propositions 1 and 2 follows immediately from Proposition 2 by setting $j=1$ in Proposition 2 and by re-arranging the Proposition 1 formula so that the $2n^{n-3}$ direct $\hat{a}_{ij(\tau)}$ are (w.l.o.g) re-labelled as the first $2n^{n-3}$ terms. See proof of Corollary to Propositions 1 and 2 in Appendix B.

Proposition 3. *The overall product of single indirect $\hat{a}_{i1(\tau)}$ terms (i.e. $\prod_{\tau=1}^{(n-2)n^{n-3}}\hat{a}_{i1(\tau)}$) can be rearranged as a product of pairs of direct terms $a_{ik}a_{k1}$, that is*

$$\prod_{\tau=1}^{(n-2)n^{n-3}}\hat{a}_{i1(\tau)}=\prod_{q=1}^{n-2}\left(a_{ik_q}a_{k_q1}\right)^{s_q}\forall i=1,2,\dots,n$$

$$\text{where}\ \sum_{q=1}^{n}s_q=(n-2)n^{n-3}$$

Comment. The proof of Proposition 3 follows from the fact that each indirect $\hat{a}_{i1(\tau)}$ corresponds to a path connecting node i to node 1 where that path is either initially of length 2 and so of the form $a_{ik}a_{k1}$ for some k, or, if longer than length 2, can be paired with its reverse path where the product of the paired indirect $\hat{a}_{i1(\tau)}$ terms reduces to a product of terms of the form $a_{ik}a_{k1}$ for some k. That is, the overall product of single indirect $\hat{a}_{i1(\tau)}$ terms $(\prod_{\tau=1}^{(n-2)n^{n-3}}\hat{a}_{i1(\tau)})$ can always be rearranged as a product of pairs of direct terms $a_{ik}a_{k1}$. See proof of Proposition 3 in Appendix B.

Proposition 4. *The number of spanning trees connecting node i to node 1 indirectly by a path of length 2 via a given node k (i.e. $i\rightarrow k\rightarrow 1$) for some $k\in\{2,\dots,n\}$where $k\neq i$, is the same $\forall k\in\{2,\dots,n\}$where $k\neq i$.*

Comment. The proof of Proposition 4 follows from symmetry in that the number of spanning trees connecting a given pair of nodes via some intermediate node must be independent of the choice of the intermediate node. See proof of Proposition 4 in Appendix B.

Corollary to Proposition 4. The number of indirect $\hat{a}_{i1(\tau)}$ in the product $\prod_{\tau=1}^{(n-2)n^{n-3}}\hat{a}_{i1(\tau)}$ of length 2 (i.e. where $\hat{a}_{i1(\tau)}=a_{ik}a_{k1}$ for some $k\in\{2,\dots,n\}$, $k\neq i$) is the same $\forall k\in\{2,\dots,n\}$ where $k\neq i$.

Comment. The Corollary to Proposition 4 follows immediately from Proposition 4 in that each indirect $\hat{a}_{i1(\tau)}$ of length 2 corresponds exactly to a path of length 2 in the underlying tree τ. See the proof of the corollary in Appendix B.

Proposition 5. *The number of pairs of spanning trees connecting node i to node 1 via a path of length >2 via k_1, $k_s\in\{2,\dots,n\}$, k_1, $k_s\neq i$, $k_1\neq k_s$ is the same $\forall\{k_1,k_s\}\subseteq\{2,\dots,n\}$, k_1, $k_s\neq i$, $k_1\neq k_s$.*

Comment. The proof of Proposition 5 follows from symmetry in that the number of pairs of spanning trees connecting a given pair of nodes via some intermediate path defined by its starting/ending nodes must be independent of the choice of these starting/ending nodes. See proof of Proposition 5 in Appendix B.

Corollary to Proposition 5. The number of indirect $\hat{a}_{i1(\tau)}$ in the product $\prod_{\tau=1}^{(n-2)n^{n-3}}\hat{a}_{i1(\tau)}$ of length >2 (i.e. where $\hat{a}_{i1(\tau)}=a_{ik_1}a_{k_1k_2}\dots a_{k_s1}$ for some $k_1,\dots,k_s\in\{2,\dots,n\}$, $k_q\neq i$ and $k_q\neq k_p\forall q\neq p$ where $q,p=1,\dots,s$) is the same for all $\{k_1,k_s\}\subseteq\{2,\dots,n\}$, k_1, $k_s\neq i$.

Comment. The Corollary to Proposition 5 follows immediately from Proposition 5 in that each indirect $\hat{a}_{i1(\tau)}$ of length >2 corresponds exactly to a path of length >2 in the underlying tree τ. See proof of the corollary in Appendix B.

3.1. Proof of the Equivalence of the RGM and GMAST preference vectors

3.1.1. Ideal-mode RGM

The expression for ideal-mode RGM is $\hat{\mathbf{u}}=\left(\hat{u}_i\right)$ where $\hat{u}_i=\left(\prod_{j=1}^{n}a_{ij}a_{j1}\right)^{\frac{1}{n}}$, as given in (4), which can be expanded as:

$$\hat{u}_i=(a_{i1}a_{11}\times a_{i2}a_{21}\times\dots\times a_{ii}a_{i1}\times\dots\times a_{in}a_{n1})^{\frac{1}{n}}\qquad(5)$$

Gathering the 1st and ith product pairs together then since $a_{ii}=a_{11}=1$, we can re-arrange as:

$$\hat{u}_i=\left((a_{i1})^2\times\prod_{q=1}^{n-2}a_{ik_q}a_{k_q1}\right)^{\frac{1}{n}}=(a_{i1})^{\frac{2}{n}}\times\left(\prod_{q=1}^{n-2}a_{ik_q}a_{k_q1}\right)^{\frac{1}{n}}\qquad(6)$$

where $k_q\in\{2,\dots,n\}$ and $k_q\neq i$ and $k_q\neq k_p\forall q\neq p$.

3.1.2. Ideal-mode GMAST

By Proposition 1, ideal-mode GMAST is $\hat{\mathbf{u}}=\left(\hat{u}_i\right)$ where $\hat{u}_i=\left(\prod_{\tau=1}^{\eta}\hat{a}_{i1(\tau)}\right)^{\frac{1}{\eta}}$ and each $\hat{a}_{i1(\tau)}$ is either

- set directly as a_{i1} (if $a_{i1}\in\mathcal{A}_\tau$), or
- derived indirectly as a transitive product of a subset of a_{ij}, say $a_{ik_1}a_{k_1k_2}\dots a_{k_s1}$, where $\left\{a_{ik_1,}a_{k_1k_2},\dots,a_{k_s1}\right\}\subseteq\mathcal{A}_\tau$(if $a_{i1}\notin\mathcal{A}_\tau$).

But by the Corollaries to Propositions 1 and 2:

$$\hat{u}_i=(a_{i1})^{\frac{2}{n}}\times\left[\prod_{\tau=1}^{(n-2)n^{n-3}}\hat{a}_{i1(\tau)}\right]^{\frac{1}{\eta}}\qquad(7)$$

And by Proposition 3, we have:

$$\left[\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)}\right]^{\frac{1}{\eta}} = \left[\prod_{q=1}^{n-2}\left(a_{ik_q}a_{k_q1}\right)^{s_q}\right]^{\frac{1}{\eta}} \quad \text{where} \quad \sum_{q=1}^{n-2} s_q = (n-2)n^{n-3} \tag{8}$$

But by the Corollaries to Propositions 4 and 5, $s_q = s\,\forall q = 1, 2, \ldots (n-2)$ and so

$$\sum_{q=1}^{n-2} s_q = (n-2)s$$

$$\therefore (n-2)s = (n-2)n^{n-3}$$

$$\therefore s = n^{n-3} \tag{9}$$

Therefore setting $\eta = n^{n-2}$ and $s_q = s = n^{n-3}$ in (8), we obtain the following

$$\left[\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)}\right]^{\frac{1}{\eta}} = \left[\prod_{q=1}^{n-2}\left(a_{ik_q}a_{k_q1}\right)^{n^{n-3}}\right]^{\frac{1}{n^{n-2}}}$$

$$= \left[\left(\prod_{q=1}^{n-2} a_{ik_q}a_{k_q1}\right)^{n^{n-3}}\right]^{\frac{1}{n^{n-2}}} \tag{10}$$

Therefore taking (7) and (10) together gives:

$$\hat{u}_i = (a_{i1})^{\frac{2}{n}} \times \left[\left(\prod_{q=1}^{n-2} a_{ik_q}a_{k_q1}\right)^{n^{n-3}}\right]^{\frac{1}{n^{n-2}}}$$

$$= (a_{i1})^{\frac{2}{n}} \times \left(\prod_{q=1}^{n-2} a_{ik_q}a_{k_q1}\right)^{\frac{1}{n}}$$

Therefore this gives:

$$\hat{u}_i = (a_{i1})^{\frac{2}{n}} \times \left(\prod_{q=1}^{n-2} a_{ik_q}a_{k_q1}\right)^{\frac{1}{n}}$$

This proves that the ideal-mode solution for RGM ($\mathbf{u}^{\mathbf{RGM}}$) is equivalent to the ideal-mode solution for GMAST ($\mathbf{u}^{\mathbf{GMAST}}$).

3.1.3. Comparing the RGM and GMAST preference vectors

Comparing the result of 3.1.1 with that of 3.1.2 proves that the ideal-mode solution for RGM ($\hat{\mathbf{u}}^{\mathbf{RGM}}$) is equivalent to the ideal-mode solution for GMAST ($\hat{\mathbf{u}}^{\mathbf{GMAST}}$).

It therefore follows that any ratio-based preference vectors for RGM and GMAST are equivalent. That is, since $\hat{\mathbf{u}}^{\mathbf{RGM}} = \hat{\mathbf{u}}^{\mathbf{GMAST}}$, then $\hat{u}_i^{RGM} = \hat{u}_i^{GMAST} \forall i = 1 \ldots n$, and therefore $(\hat{u}_i/\hat{u}_j)^{RGM} = (\hat{u}_i/\hat{u}_j)^{GMAST} \forall i = 1 \ldots n$. And since $\hat{u}_i/\hat{u}_j = \hat{w}_i/\hat{w}_j$ for any ratio-based preference vector (and therefore for both RGM and GMAST), then $(\hat{w}_i/\hat{w}_j)^{RGM} = (\hat{w}_i/\hat{w}_j)^{GMAST} \forall i = 1 \ldots n$.

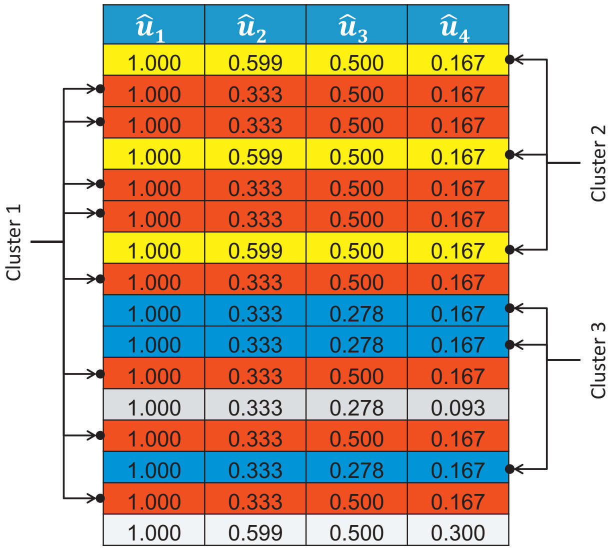
This proves that any ratio-based preference vectors obtained from RGM and GMAST are equivalent for a PC matrix having a complete set of judgements.

4. Illustrative examples for GMAST

4.1. Inconsistency analysis

While the focus of this paper has been on the mathematical equivalence of the preference vectors for GMAST and RGM, we have also mentioned some of the additional benefits of using the GMAST approach, in particular, the ability to facilitate inconsistency analysis and the inherent ability to handle incomplete PC matrices. We discuss the former here in this section and the latter in Section 4.2 below.

Table 1
The “spanning trees” solutions for A_{ex} (in ideal-mode).


Cluster 1
Cluster 2
Cluster 3

$\hat{u}_1$	$\hat{u}_2$	$\hat{u}_3$	$\hat{u}_4$
1.000	0.599	0.500	0.167
1.000	0.333	0.500	0.167
1.000	0.333	0.500	0.167
1.000	0.599	0.500	0.167
1.000	0.333	0.500	0.167
1.000	0.333	0.500	0.167
1.000	0.599	0.500	0.167
1.000	0.333	0.500	0.167
1.000	0.333	0.278	0.167
1.000	0.333	0.278	0.167
1.000	0.333	0.500	0.167
1.000	0.333	0.278	0.093
1.000	0.333	0.500	0.167
1.000	0.333	0.278	0.167
1.000	0.333	0.500	0.167
1.000	0.599	0.500	0.300

There are a number of ways in which GMAST can facilitate inconsistency analysis and we illustrate these with the help of the 4 × 4 PC matrix example taken from Hartvigsen (2005) i.e.

$$A_{ex} = \begin{bmatrix} 1 & 3 & 2 & 6 \\ 1/3 & 1 & 6/5 & 2 \\ 1/2 & 5/6 & 1 & 3 \\ 1/6 & 1/2 & 1/3 & 1 \end{bmatrix}$$

The CR value for this matrix is 0.016 which falls well below the widely-accepted threshold of 0.1, and so, the PC matrix is deemed eligible for using the REV method to calculate the preference vector. However, as discussed by Hartvigsen (2005), in this example, the preference vector obtained from the REV method gives $w_1 > w_3 > w_2 > w_4$ which is not the correct order of preference (since with a little thought it can easily be seen that the correct order should be $w_1 > w_2 > w_3 > w_4$). In fact, all the existing methods were shown to have produced incorrect preference orders (also by Hartvigsen, 2005). In this scenario, the value of CR = 0.016 is therefore at best misleading in terms of its assessment of acceptable inconsistency. However, with the help of the spanning trees approach, we can generate the set of 16 possible (ideal-mode) preference vectors directly from the DM’s comparison judgements, and analyse these for inconsistency.

For example, we can choose to cluster them together according to their similarity/dissimilarity, as shown in Table 1, where three clusters of preference vectors are clearly evident. The largest cluster (Cluster 1) shows that on the one hand, the DM prefers Element 3 over Element 2 whereas another cluster (i.e. Cluster 2) shows that the DM seems to prefer Element 2 over Element 3. In other words, the DM is in at least two minds, and the critical issue is with regards to his order of preference for Elements 2 and 3. The spanning trees approach has uncovered this “two mindedness” while the other methods failed to highlight this issue and not only this but the spanning trees approach has identified where the “two mindedness” arises from.

Of course, there exist other ways to present this information on inconsistency to the DM, such as, using a dimensionality reduction technique like principal component analysis for better visualization. For example, Fig. 1 shows the 16 preference vectors derived from A_{ex} using the first two principal components (on a

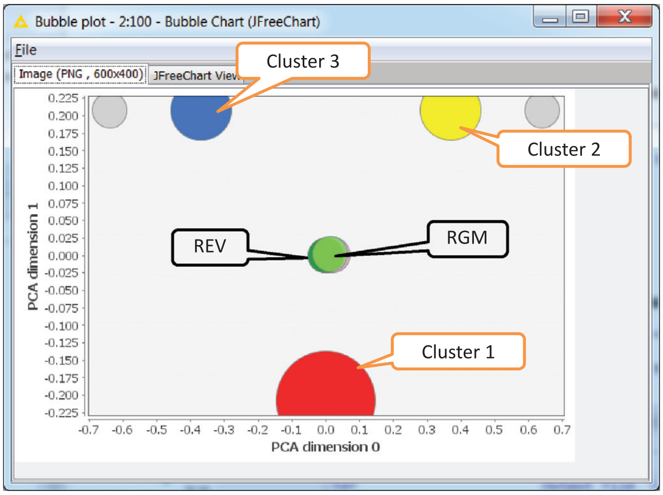


Fig. 1. Visualising the “spanning trees” solutions for A_{ex} on the first two principal components axes.

logarithmic scale). Note that in this example, each cluster happens to contain a set of identical preference vectors, and therefore, the size of each bubble is proportional to the number of preference vectors contained within the cluster. Using this representation, it is easy to see that the single preference vector obtained from other methods (e.g. REV and RGM) usually represents a compromise and is not necessarily representative of any of the clusters, and therefore is not necessarily representative of any of the DM’s actual mindsets.

We consider this inconsistency analysis to be an area of future research that has the potential to provide the DM with an aid to revise their judgements, or to select a more appropriate solution interactively, as well as to propose a new way of measuring inconsistency.

4.2. Incomplete matrices

As asserted earlier, we contend that GMAST is inherently applicable to incomplete PC matrices while the RGM method is not and instead requires estimation/imputation of the missing elements in the PC matrix as a preliminary step.

The reasoning behind this assertion is not immediately obvious since it is clearly theoretically possible to calculate the row geometric mean of the non-missing elements of any given row of a matrix and so, at first glance, it might seem that there is no reason why we should not adopt this ‘non-missing’ approach. However with the help of the example below we show that this ‘non-missing’ approach is a fundamentally flawed procedure in that it can lead to bizarre results.

For example, considering the following incomplete PC matrix:

$$A'_{ex} = \begin{bmatrix} 1 & 3 & 2 & 6 \\ 1/3 & 1 & - & 2 \\ 1/2 & - & 1 & 3 \\ 1/6 & 1/2 & 1/3 & 1 \end{bmatrix}$$

which we have carefully constructed from Hartvigsen (2005) (as discussed in the previous sub-section) in such a way that the matrix remains (perfectly) consistent although it has a missing judgement a_{23}.

Obviously, as with any case of a consistent set of preferences, there exists an ideal preference vector and any prioritization method must be able to produce this ideal preference vector. However, we show below that while GMAST does find this ideal vector, the RGM approach does not. In other words, the RGM approach is deficient in its ability to guarantee always finding the ideal preference vector for a consistent but incomplete set of preferences.That is, we can see that the judgement a_{23} (and therefore a_{32}) is missing in the matrix, but yet the ideal prioritization clearly exists due to the fact that all the non-missing elements in the matrix are consistent with each other.

Table 2
The “spanning trees” solutions for A_{ex} (in ideal-mode).

	$\hat{u}_1$	$\hat{u}_2$	$\hat{u}_3$	$\hat{u}_4$
Cluster 1	1.000	0.333	0.500	0.167
	1.000	0.333	0.500	0.167
	1.000	0.333	0.500	0.167
	1.000	0.333	0.500	0.167
	1.000	0.333	0.500	0.167
	1.000	0.333	0.500	0.167
	1.000	0.333	0.500	0.167
	1.000	0.333	0.500	0.167

The number of spanning trees is obviously reduced as each and every one of the trees that span the missing elements are now absent. This can be visualized in Table 2 which shows a subset of solutions generated by the available spanning trees, and which is essentially a subset of Table 1. The remaining trees are still consistent with each other and generate the same preference vector. Therefore, the ’non-missing’ GMAST will still produce the ideal preference vector.

The spanning tree analysis has interestingly made it obvious that all the “spanning-tree” solutions are identical for this incomplete matrix, and are equal to (1, 0.333, 0.5, 0.167) in ideal-mode.

However, applying ’non-missing’ RGM to A'_{ex}, that is taking the geometric mean of the non-missing elements in each row, gives:

$$RGM(A'_{ex}) = \begin{bmatrix} (1 \times 3 \times 2 \times 6)^{1/4} \\ (1/3 \times 1 \times 2)^{1/3} \\ (1/3 \times 1 \times 3)^{1/3} \\ (1/6 \times 1/2 \times 1/3 \times 1)^{1/4} \end{bmatrix} = \begin{bmatrix} 2.4495 \\ 0.8736 \\ 1.1447 \\ 0.4082 \end{bmatrix}$$

which can be represented in ideal-mode as $w = \begin{bmatrix} 1 \; 0.357 \; 0.4670.167 \end{bmatrix}$. This is clearly not identical to the ideal preference vector and so does not adhere to the DM’s judgements.

Having said this, if we impute the missing judgement as $a_{23} = a_{21}a_{13} = 2/3$ (or $a_{23} = a_{24}a_{43} = 2/3$) , the (complete) RGM approach will provide the following results:

$$RGM(A'_{ex}) = \begin{bmatrix} (1 \times 3 \times 2 \times 6)^{1/4} \\ (1/3 \times 1 \times 2/3 \times 2)^{1/3} \\ (1/3 \times 2/3 \times 1 \times 3)^{1/3} \\ (1/6 \times 1/2 \times 1/3 \times 1)^{1/4} \end{bmatrix} = \begin{bmatrix} 2.4495 \\ 0.8165 \\ 1.2247 \\ 0.4082 \end{bmatrix}$$

which gives us the ideal-mode preference vector $w = \begin{bmatrix} 1 \; 0.333 \; 0.5 \; 0.167 \end{bmatrix}$ but which does strictly adhere to the DM’s provided judgements. That is, the example shows that applying the RGM operation directly to incomplete PC matrices without estimating the missing judgements is fundamentally flawed.

So far, we have discussed the situation in which the PC matrix is incomplete yet consistent, which we consider sufficient enough to highlight the shortcomings of the RGM method in so far as its ability to handle incomplete PC matrices. However, note that in the presence of an incomplete yet inconsistent PC matrix, the situation becomes more complex due to the fact that no ideal preference vector exists and so it is no longer merely a matter of checking to see whether or not this ideal vector has indeed been found by the prioritization method. Instead, we use several different criteria to relatively assess the strength of these methods. We discuss these criteria in the next section below.

5. Comparative analysis of prioritization methods

As mentioned earlier, the well-known performance criteria for comparing prioritization methods are: minimal deviation from the DM's judgements, computational complexity, ability to handle incomplete sets of judgements, adherence to geometric properties, and ability to measure inconsistency. As also mentioned earlier, we contend that a method must also be able to facilitate inconsistency analysis rather than just measure it.

Focusing on the minimal deviation from the DM's judgements, there exist two types of deviations i.e. cardinal and ordinal, and both have been well investigated (Golany & Kress, 1993; Siraj, Mikhailov, & Keane, 2015). The optimization methods naturally outperform other methods with respect to their own (cardinal) objective error functions, however, as reported in Siraj et al. (2012b), the DLS method does not perform well in the ordinal deviation criterion. Interestingly, the REV and RGM methods perform satisfactorily well in both the cardinal and ordinal deviations criteria. Since the GMAST preference vector is mathematically equivalent to the RGM preference vector, it also performs satisfactorily with respect to the cardinal and ordinal deviations.

In terms of the computational complexity, the RGM method is arguably the most straightforward process i.e. taking the geometric mean of all the values in each row of the PC matrix. By contrast, the optimization methods are relatively more complex as they depend upon the optimization method used (e.g. simplex method, simulated annealing, genetic algorithm etc.) and their selection of parameters (e.g. number of iterations, acceptable threshold of error, etc.). The REV method depends upon the eigenvector calculation process which is relatively simpler than the optimization methods but not as straightforward as the RGM method. The complexity of the GMAST method varies with the value of n, where if all the spanning trees are to be enumerated then the number of operations are acceptable for $n < 9$, i.e. within Miller's definition of 7 ± 2 as the limit on a DM's capacity for comparing elements (Miller, 1956). And with partial enumeration, the number of operations are acceptable for $n > 9$.

Considering the ability of handling incomplete sets of judgements, the REV and RGM methods have no inherent capability to deal with this issue and so they need to have the missing judgements estimated with in a preliminary step (Ergu et al., 2011; Harker, 1987b). However, as with the optimization methods (e.g. LLS, DLS, and LLAV), the GMAST method is able to obtain a preference vector without estimating the missing judgements. This is an important benefit due to the fact that PC matrices are often incomplete - indeed, the probability of acquiring an incomplete set of comparison judgements from a DM increases with n (Harker, 1987a).

Considering the adherence to the geometric properties, as described by Barzilai (1997), the RGM method is the only existing method that strictly adheres to the geometric properties including the independence of scale inversion and the independence of order of operations. Since the GMAST preference vector is mathematically equivalent to the RGM preference vector, the GMAST method satisfies these geometric properties also.

With regards to measuring inconsistency, the widely used method of REV proposes the Consistency Ratio (CR) and uses it along with a threshold to accept or reject the DM's provided comparison judgements as suitable or not for preference vector development. Similarly, the Geometric Consistency Index (GCI) has been introduced for the RGM method with the same purpose of accepting/rejecting the DM's judgements. However, this is only part of the problem as although the existing methods provide a measure of inconsistency, they do not attempt to facilitate inconsistency analysis. GMAST, on the other hand, does have this ability since the "spanning trees" preference vectors can be analysed statistically to detect the impact of inconsistency on the variability in the final solution - for example, finding the clusters of similar preference vectors and performing inter-cluster and intra-cluster analysis.

The comparative analysis of the prioritization methods can be summarized in Table 3 by evaluating each of them across the six performance criteria which should be considered when choosing a method. The use of label '✔' implies that the method is highly suitable, '✓' that the method is acceptable, while '✗' implies that the method performs poorly on the given criterion. For example, considering the first row "Minimal deviation from DM judgements", we can comfortably state that all methods perform well due to the fact that each method has its own criterion for minimal deviation. However, this "equivalence of performance" is not necessarily the case when comparing the methods across the other criteria. For example, considering the second row "Minimal number of ordinal violations", although most methods have performed equally well, DLS does not. Furthermore, even when a method performs well in one criterion, it tends to perform less well on another. For example, considering the third row "Acceptable computation time", we see that REV and RGM perform well but yet cannot be used for incomplete sets of judgements without some preliminary process (as shown in the fourth row).

Continuing in this way, and considering the fifth row "Adhering to the geometric properties", we see that LLS and RGM are two of the three methods which satisfy these geometric properties, yet neither of the methods has the ability to facilitate inconsistency analysis. That is, we see that each method performs well on some criteria but tends to fall short of adequacy on other criteria, apart from GMAST which performs well on all criteria.

It can therefore be concluded that although no method consistently outperforms all of the other methods, the GMAST method is the only method which is better (or equally good) than all the rest across all the six criteria.

6. Applications of spanning tree analysis

6.1. Example

In addition to the illustrative example provided in Section 4.1, we develop the inconsistency analysis ideas further here using the famous school selection example, first discussed by Saaty and Rogers (1976), where six criteria of "Learning", "Friends", "School life", "Vocational training", "College preparation", and "Music classes" were shortlisted for assessing the available schools. These criteria were compared in a pairwise fashion as given below:

$$\mathbf{A}_{\textbf{criteria}} = \begin{bmatrix} 1 & 4 & 3 & 1 & 3 & 4 \\ 1/4 & 1 & 7 & 3 & 1/5 & 1 \\ 1/3 & 1/7 & 1 & 1/5 & 1/5 & 1/6 \\ 1 & 1/3 & 5 & 1 & 1 & 1/3 \\ 1/3 & 5 & 5 & 1 & 1 & 3 \\ 1/4 & 1 & 6 & 3 & 1/3 & 1 \end{bmatrix}$$

The arrangement of columns is in the same order as the criteria were introduced above. The level of inconsistency in this PC matrix

Table 3
Comparison of a number of prioritization methods

	REV	LLS	RGM	DLS	LLAV	GMAST
Minimal deviation from DM judgements	✓	✓	✓	✓	✓	✓
Minimal number of ordinal violations	✓	✓	✓	✗	✓	✓
Acceptable computation time	✓	✓	✓	✓	✓	✓
Handling incomplete set of judgements	✗	✓	✗	✓	✓	✓
Adherence to the geometric properties	✗	✓	✓	✗	✓	✓
Ability to measure inconsistency	✓	✓	✓	✗	✗	✓
Ability to facilitate inconsistency analysis	✗	✗	✗	✗	✗	✓

was calculated using CR which turned out to be quite high (i.e. 0.24). However, the matrix was still considered for further analysis, and the following preference vector was obtained using the REV method (values are rounded up to two decimal places):

$$w_{EV} = \left[0.32\ 0.14\ 0.03\ 0.13\ 0.24\ 0.14\right]$$

The complete problem involved the selection of one out of three schools by assessing them against each of the six criteria and coming up with an overall ranking based on their performances in these six criteria (see Saaty and Rogers, 1976 for complete details). Eventually, the authors mention that “The son went to school A because it had almost the same evaluation as school B”. The PC matrix $\mathbf{A}_{\mathbf{criteria}}$ can be used to generate 1296 (i.e. $n^{n-2} = 6^4$) trees and their respective preference vectors. Due to a large number of preference vectors, we refrain from listing them all here due to the limitations of space, however, we can visualize these vectors using the dimensionality reduction technique (as discussed before in Section 4) by plotting each vector as a point positioned according to its projections on the first few principal components. In this way, we can capture as much variance as possible in the top three components (first and second components are shown in Figure 2a while the first and third components are shown in Fig. 2b). However, when the location of these points overlap, we represent this multiplicity by increasing the size of the point, as shown in these figures in the form of bubble charts. The size of each bubble represents the number of preference vectors lying at that location.

6.2. Analysis

Fig. 2 is indeed an interesting visual representation revealing all of the preference vectors that can be obtained directly from the DM’s comparative judgements. In other words, each and every one of these preference vectors is a direct reflection of what the DM has told us i.e. with no aggregation or adjustments (unlike the REV or RGM preference vectors). However, as stated earlier, there is nothing to stop the DM (or analyst) from choosing to aggregate all of these preference vectors e.g. by taking their geometric mean, which according to the proof in Section 3, turns out of course to be identical to the RGM (notice the RGM solution lying in the middle of all the “spanning tree” preference vectors in the two graphs).

Note that the figure also shows that the REV and RGM preference vectors lie close to each other, however, this leads us to a different discussion which is out of the scope of this article (see Siraj et al., 2012b for details).

Just replicating what we might otherwise have produced from RGM is not of course the point. Rather we want to exploit the additional information that GMAST has provided over and above the generation of the single GMAST/RGM preference vector and we discuss a number of ideas below.

6.2.1. Closest representative solution

Notice in Fig. 2 that REV and GMAST/RGM have all produced preference vectors that lie somewhere in the middle of the set of generated “spanning tree” preference vectors, and so in this sense, all are an average representation of the actual preferences provided by the DM in terms of comparative judgements. All of these preference vectors are artificial to a degree - and so in this sense are not specific representations of any of the actual preferences provided by the DM. However, with the help of the spanning tree analysis, we could choose to identify the “spanning tree” solution that lies closest to the GMAST/RGM preference vector (e.g. using Euclidean distance). The idea being to choose a preference vector that is representative in both of the senses described above - that is which is both an average representation and a specific representation of the DM’s preferences and so because of the latter can be thought of as a realization of the average representation. This proposition needs more investigation for its empirical validity and is considered an area of future work.

6.2.2. Subset of trees

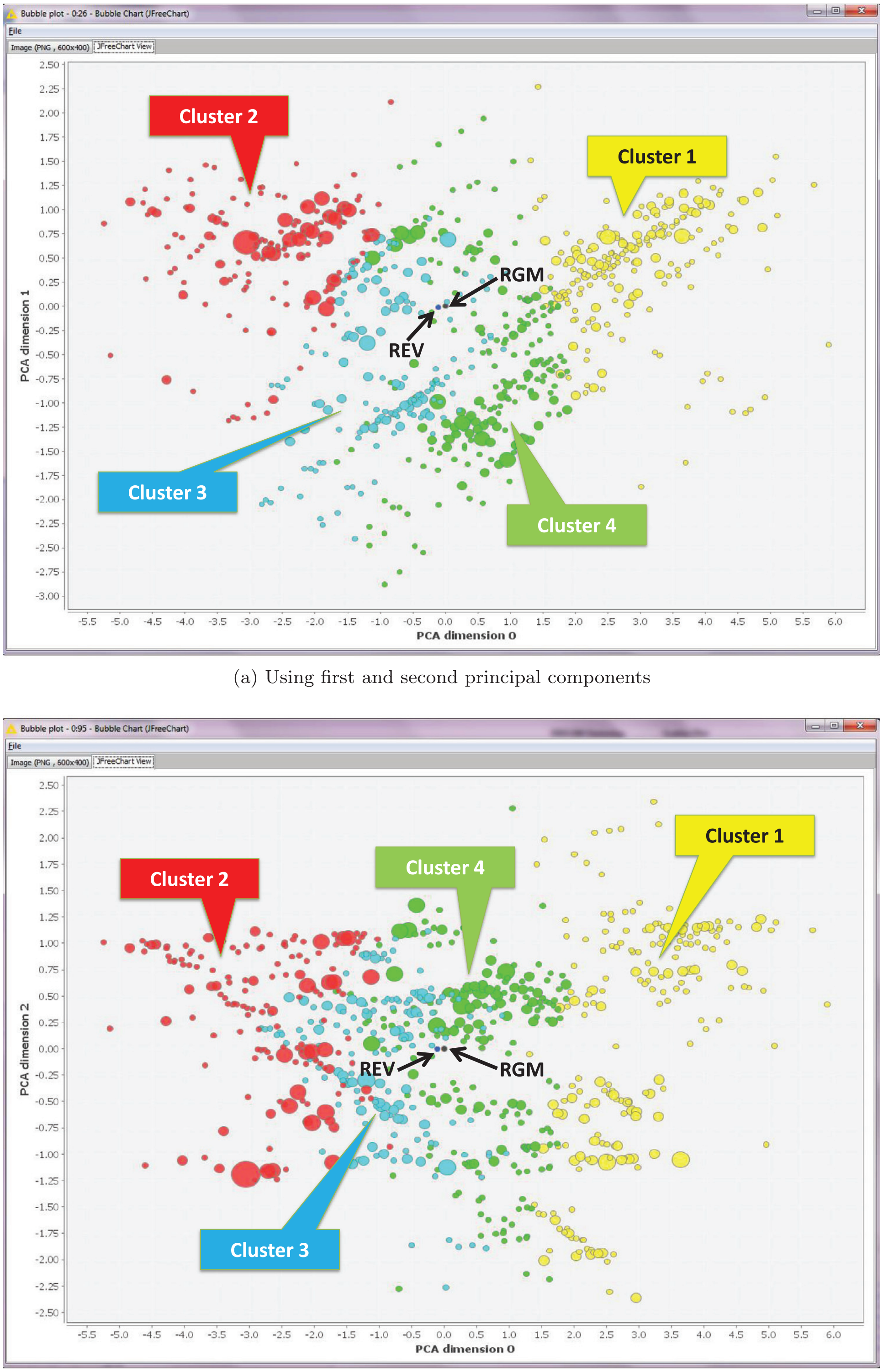
Alternatively, the DM (or analyst) may like to use the “spanning-trees” preference vectors to gain more insights into the comparison judgements and preference vectors, for example, by clustering the “spanning tree” preference vectors with respect to a distance measure. With the result that these clusters could be offered to the DM as a visual aid so that the DM could choose to accept or reject some of these clusters by inspection, and possibly, choose the geometric mean of the filtered “spanning trees” preference vectors.

In Fig. 2, we have shown the possibility of creating four clusters using the K-Means algorithm with Euclidean distance, n.b. the use of K-Means is only for demonstration purposes; the number of clusters and the choice of clustering algorithm are both subject to further investigation. For example, one may question the use of K-Means as we have no a priori information about the number of clusters present in the “spanning trees” data; and therefore, may seek for some other algorithm like DBSCAN (i.e. density-based scanning) where the number of clusters are not required as an input.

6.2.3. Judgements revision

Another interesting use of the spanning trees analysis is to provide an interactive aid to the DM in revising his/her comparative judgements. Since each judgement in the PC matrix contributes to a certain number of trees, we can highlight the trees for a single judgement (interactively chosen by the DM). In this way, the DM can locate the preference vectors affected by one of his/her judgements, and therefore, have the possibility of revising the judgements if required.

We have discussed these applications to show the benefits of using the spanning trees analysis for facilitation of inconsistency analysis, however, these applications need to be further developed as areas of future research.



(a) Using first and second principal components

(b) Using first and third principal components

Fig. 2. Visualising the “spanning trees” preference vectors for the school example.

7. Conclusions

We have discussed the “spanning tree” method (i.e. GMAST) and have proved the mathematical equivalence of its preference vector to that of the RGM approach while highlighting the additional benefits of the GMAST method in its entirety. That is, we have identified an approach for generating a preference vector which has the mathematical properties of RGM, and yet, is inherently applicable to incomplete pairwise comparison matrices and also facilitates the use of statistical techniques to gain insights into inconsistency. That is, in its entirety, the GMAST method has more to offer than other prioritization methods.

This opens up several interesting avenues for further research into the use of statistical approaches for pairwise comparisons.

For example, this creates an opportunity to examine the similarity/dissimilarity within the set of "spanning tree" preference vectors (possibly clustering them together based on some *distance* measure), or using a democratic approach to select the most preferred solution, or to present clusters of preference vectors as a visual aid to the DM for interactive inconsistency resolution. Moreover, we have discussed the criteria matrix from the school example, however, the same analysis can also be carried out for all the PC matrices in a given multicriteria decision problem. For example, we can generate the "spanning-tree" preference vectors from the PC matrices comparing schools under each of the criteria. In this case, we have 1296 preference vectors as criteria preference vectors, and for each of these vectors, we can calculate scores for each school using the PC matrices comparing schools. Generalizing this to m options/alternatives and n criteria, we can calculate the number of all possible preference vectors (i.e. overall preference vectors for a given problem) for the m alternatives as below:

Number of "spanning-trees" preference vectors for criteria PC matrix $= n^{n-2}$

Number of PC matrices under the criteria $= n$

Number of "spanning-trees" preference vectors for options PC matrix $= m^{m-2}$

Number of "spanning-trees" preference vectors for single vector from criteria PC matrix $= (m^{m-2})^n$

Total number of possible preference vectors $= n^{n-2}m^{n(m-2)}$

In the school example, this expression means that the total number of possible preference vectors is 944,784. Considering this expression, one may argue that the number of preference vectors grows exponentially and the problem may become intractable very quickly. Nonetheless, we propose to use stochastic analysis in such cases when generating the complete set of preference vectors is impractical. We consider this an interesting new area in preference elicitation which enables the DMs to visualize a huge number of possible preference vectors, all of which directly emerge from the DM's given set of comparative judgements.

Appendix A. Table of Notation

Symbol	Description
n	number of elements evaluated using pairwise comparison judgements
a_{ij}	ratio judgement comparing ith element to jth element
A	symmetrically reciprocal matrix constructed from the given set of a_{ij}
$\mathbf{w}$	preference vector (or preference weight vector)
w_i	preference score (or preference weight) of ith element
η	total number of spanning trees possible for a given A (where $\eta = n^{n-2}$)
τ	a spanning tree
$\mathcal{E}_\tau$	set of edges $[i \to j]$ in a spanning tree τ (a subset of edges from a fully-connected graph)
$e_{ij(\tau)}$	a boolean variable which is 1 when the edge $[i \to j] \in \mathcal{E}_\tau$, otherwise 0
$\mathcal{A}_\tau$	a sufficient subset of $(n-1)$ of the $\frac{n(n-1)}{2}$ elements in the lowers triangle of A
$\hat{A}_\tau$	a consistent PC matrix constructed from the subset $\mathcal{A}_\tau$
$\hat{a}_{ij(\tau)}$	an entry in the consistent matrix $\hat{A}_\tau$
$\hat{\mathbf{w}}$	estimated preference vector (or estimated preference weight vector)
$\hat{\mathbf{u}}$	estimated normalized preference vector in an ideal-mode
$\hat{u}_i$	estimated normalized score (or weight) for ith element of $\hat{\mathbf{u}}$
$\hat{\mathbf{w}}_\tau$	estimated $\hat{\mathbf{w}}$ from the spanning tree τ
$\hat{\mathbf{u}}_\tau$	estimated $\hat{\mathbf{u}}$ from the spanning tree τ
$\hat{u}_{i(\tau)}$	ith elements of $\hat{\mathbf{u}}_\tau$

Appendix B

Proof of Proposition 1. Since $\hat{A}_\tau$ is a PC matrix it can be used to derive a $\hat{\mathbf{w}}_\tau$, and since $\hat{A}_\tau$ is consistent by construction, $\hat{\mathbf{w}}_\tau$ can be derived from any column of $\hat{A}_\tau$, and choosing column 1 means that $\hat{\mathbf{w}}_\tau$ is in ideal-mode $\hat{\mathbf{u}}_\tau$ since column 1 is $\left[1, \hat{a}_{21(\tau)}, \ldots, \hat{a}_{n1(\tau)}\right]^T$

i.e. $\hat{\mathbf{u}}_\tau = \left(\hat{u}_{i(\tau)}\right)$ where $\hat{u}_{i(\tau)} = \hat{a}_{i1(\tau)}$, and so $\hat{\mathbf{u}} = \left(\hat{u}_i\right)$ where

$$\hat{u}_i = \left(\prod_{\tau=1}^{\eta} \hat{u}_{i(\tau)}\right)^{\frac{1}{\eta}} = \left(\prod_{\tau=1}^{\eta} \hat{a}_{i1(\tau)}\right)^{\frac{1}{\eta}}$$

And from Section 2.1 with $j = 1$ each $\hat{a}_{i1(\tau)}$ is either

- set directly as a_{i1} (if $a_{i1} \in \mathcal{A}_\tau$), or
- derived indirectly as a transitive product of a subset of a_{ij}, say $a_{ik_1}a_{k_1k_2}\ldots a_{k_s1}$, where $\left\{a_{ik_1}, a_{k_1k_2}, \ldots, a_{k_s1}\right\} \subseteq \mathcal{A}_\tau$(if $a_{i1} \notin \mathcal{A}_\tau$). □

Proof of Proposition 2. By the fundamentals of the "spanning tree" approach (see Section 2.1), each $\hat{a}_{ij(\tau)}$ in $\hat{A}_\tau$ is set directly as a_{ij} if $a_{ij} \in \mathcal{A}_\tau$. But also $a_{ij} \in \mathcal{A}_\tau \iff [i \to j] \in \mathcal{E}_\tau$ i.e. the number of $\hat{A}_\tau$ where $\hat{a}_{i1(\tau)}$ is set directly as a_{ij} is the number of spanning trees which contain the edge $[i \to j]$.

Let

$$e_{ij(\tau)} = \begin{cases} 1 & \text{if } [i \to j] \in \mathcal{E}_\tau \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

then summing $e_{ij(\tau)}$ over all spanning trees, $\sum_{\tau=1}^{\eta} e_{ij(\tau)}$ must be the number of spanning trees which contains the edge $[i \to j]$, and so summing $\sum_{\tau=1}^{\eta} e_{ij(\tau)}$ over all edges means that $\sum_{i>j}\sum_{\tau=1}^{\eta} e_{ij(\tau)}$ is the total number of occurrences of all edges $[i \to j]$ over all spanning trees (counting multiple occurrences of the same $[i \to j]$ separately).

But reversing the order of summation gives

$$\begin{aligned} \sum_{i>j}\sum_{\tau=1}^{\eta} e_{ij(\tau)} &= \sum_{\tau=1}^{\eta}\sum_{i>j} e_{ij(\tau)} \\ &= \sum_{\tau=1}^{\eta}(n-1) \text{ since } \sum_{i>j} e_{ij(\tau)} \text{ is the number of edges} \\ &\qquad \text{in tree } \tau \text{ which by definition is } (n-1) \\ &= \eta(n-1) \\ &= (n-1)n^{n-2} \end{aligned} \tag{12}$$

Also since by symmetry $\sum_{\tau}^{\eta} e_{ij(\tau)} = C$ for some constant C for each and every edge $[i \to j]$ (since each edge occurs in the same number of trees).

Then

$$\begin{aligned} \sum_{i>j}\sum_{\tau=1}^{\eta} e_{ij(\tau)} &= \sum_{i>j} C \\ &= \frac{n(n-1)}{2} C \text{ (since there are} \\ &\quad \frac{n(n-1)}{2} \text{ unique edges in total)} \end{aligned} \tag{13}$$

Equating (12) and (13), we have

$$\begin{aligned} \frac{n(n-1)}{2} C &= (n-1)n^{n-2} \\ C &= \frac{2(n-1)n^{n-2}}{n(n-1)} \\ C &= 2n^{n-3} \end{aligned}$$

Therefore, there are $2n^{n-3}$ trees in which $[i \to j] \in \mathcal{E}_\tau$ and so there are $2n^{n-3}$ $\mathcal{A}_\tau$ where $a_{ij} \in \mathcal{A}_\tau$ and so there are $2n^{n-3}$ $\hat{A}_\tau$

where $\hat{a}_{ij(\tau)}$ is set directly as a_{ij}. And since there are n^{n-2} trees in total, there are n^{n-2} $\mathcal{A}_\tau$ in total and so n^{n-2} $\hat{A}_\tau$ in total and so there are $n^{n-2} - 2n^{n-3} = (n-2)n^{n-3}\hat{A}_\tau$ where $\hat{a}_{i1(\tau)}$ is derived indirectly. □

Proof of Corollary to Propositions 1 and 2. By Proposition 2 with $j = 1$, then w.l.o.g. gathering all $2n^{n-3}$ direct $\hat{a}_{i1(\tau)}$ together and labelling the $(n-2)n^{n-3}$ trees in which $\hat{a}_{i1(\tau)}$ is derived indirectly as $\tau = 1, 2, ..., (n-2)n^{n-3}$. Then

$$\hat{u}_i = \left[(a_{i1})^{2n^{n-3}} \left(\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)} \right) \right]^{\frac{1}{\eta}}$$

$$= \left[(a_{i1})^{2n^{n-3}} \right]^{\frac{1}{n^{n-2}}} \left[\left(\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)} \right) \right]^{\frac{1}{\eta}}$$

$$= (a_{i1})^{\frac{2}{n}} \left[\left(\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)} \right) \right]^{\frac{1}{\eta}}$$

□

Proof of Proposition 3. By definition each indirect $\hat{a}_{i1(\tau)}$ is derived from a product of a subset of $a_{ij} \subseteq \mathcal{A}_\tau$.

So either

a) $\hat{a}_{i1(\tau)}$ is of "length 2" i.e.

$$\hat{a}_{i1(\tau)} = a_{ik}a_{k1} \tag{14}$$

for some $k \in \{2, ..., n\}$ and $k \neq i$ and the proof is complete. (Note $k \neq i$, 1 since if $k = i$ or 1, then $\hat{a}_{i1(\tau)} = a_{i1}$ and so $\hat{a}_{i1(\tau)}$ is direct but by definition $\hat{a}_{i1(\tau)}$ is indirect).

OR

b) $\hat{a}_{i1(\tau)}$ is of "length > 2" i.e.

$$\hat{a}_{i1(\tau)} = a_{ik_1}a_{k_1k_2}a_{k_2k_3}...a_{k_{s-1}k_s}a_{k_s1} \tag{15}$$

for some $k_1...k_s \in \{2, ..., n\}$, $k_q \neq i \forall q$, $s \geq 2$ and $k_q \neq k_p \forall p \neq q$.

But $a_{ik_1}a_{k_1k_2}a_{k_2k_3}...a_{k_{s-1}k_s}a_{k_s1}$ corresponds to the path $i \to k_1 \to k_2 \to ... \to k_{s-1} \to k_s \to 1$ within a spanning tree τ connecting i to 1. And by reversing this path in τ (i.e. forming $i \to k_s \to k_{s-1} \to ... \to k_2 \to k_1 \to 1$) but keeping all other paths in τ unchanged, we can create another spanning tree τ' where $\hat{a}_{i1(\tau')}$ is also of "length>2" and is of the form $a_{ik_s}a_{k_sk_{s-1}}...a_{k_3k_2}a_{k_2k_1}a_{k_11}$ so that pairing τ with τ' means that:

$$\hat{a}_{i1(\tau)} \times \hat{a}_{i1(\tau')} = \left(a_{ik_1}a_{k_1k_2}a_{k_2k_3}...a_{k_{s-1}k_s}a_{k_s1}\right)\left(a_{ik_s}a_{k_sk_{s-1}}...a_{k_3k_2}a_{k_2k_1}a_{k_11}\right)$$

And rearranging this expression by pairing like terms together gives:

$$\hat{a}_{i1(\tau)} \times \hat{a}_{i1(\tau')} = \left(a_{ik_1}a_{k_s1}\right)\left[\left(a_{k_1k_2}a_{k_2k_1}\right)\left(a_{k_2k_3}a_{k_3k_2}\right)...\left(a_{k_{s-1}k_s}a_{k_sk_{s-1}}\right)\right] \left(a_{ik_s}a_{k_11}\right)$$

which by reciprocity:

$$= \left(a_{ik_1}a_{k_s1}\right)\left[\left(a_{k_1k_2}\frac{1}{a_{k_1k_2}}\right)\left(a_{k_2k_3}\frac{1}{a_{k_2k_3}}\right)... \left(a_{k_{s-1}k_s}\frac{1}{a_{k_{s-1}k_s}}\right)\right]\left(a_{ik_s}a_{k_11}\right) \tag{16}$$

$$= a_{ik_1}a_{k_s1} \times 1 \times ... \times 1 \times a_{ik_s}a_{k_1th1}$$

$$= \left(a_{ik_1}a_{k_11}\right) \times \left(a_{ik_s}a_{k_s1}\right) \tag{17}$$

Therefore, each $\hat{a}_{i1(\tau)}$ estimated through a path of length greater than 2 can be paired with another $\hat{a}_{i1(\tau')}$ estimated through another path of length greater than 2, and more interestingly, the product pair of the form $\hat{a}_{i1(\tau)} \times \hat{a}_{i1(\tau')}$ can be reduced to a product pair of the form $\left(a_{ik_1}a_{k_11}\right) \times \left(a_{ik_s}a_{k_s1}\right)$. That is, taking (14) and (17) together, the product of the terms of the form $\hat{a}_{i1(\tau)}$ can be reduced to a product of the terms of the form $a_{ik}a_{k1}$ with the number of terms in the product $\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)}$ preserved, so that

$$\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)} = \prod_{q=1}^{n-2} \left(a_{ik_q}a_{k_q1}\right)^{s_q}$$

where s_q is the total number of occurrences of $a_{ik_q}a_{k_q1}$ in all the trees in which $\hat{a}_{i1(\tau)}$ is derived indirectly and where $\sum_{q=1}^{n-2} s_q = (n-2)n^{n-3}$. □

Proof of Proposition 4. Let

τ = a spanning tree connecting node i to node 1 indirectly by a path of length 2,

$\mathcal{T}_k = \{\tau | \tau$ connects node i to node 1 via a path $i \to k \to 1$ for some $k \in \{2, ..., n\}$ where $k \neq i\}$

n_k = number of τ in $\mathcal{T}_k$

Then if k_1 and $k'_1 \in \{2, ..., n\}$ and $k_1 \neq k'_1$ and $k_1, k'_1 \neq i$ and if $\tau \in \mathcal{T}_{k_1}$, then τ connects node i to node 1 indirectly via a path $i \to k_1 \to 1$ and as τ is a spanning tree, k'_1 must be connected within τ.

But by reversing the roles of k_1 and k'_1 in τ (keeping all else unchanged) we can create a spanning tree τ' in which node i is connected to node 1 indirectly via a path $i \to k'_1 \to 1$, and so $\tau' \in \mathcal{T}_{k'_1}$.

That is, if $k_1, k'_1 \in \{2, ..., n\}$ and $k_1 \neq k'_1$ and $k_1, k'_1 \neq i$, then for each $\tau \in \mathcal{T}_{k_1} \exists$ a $\tau' \in \mathcal{T}_{k'_1}$, and so $n_{k_1} \leqslant n_{k'_1}$.

But using the same argument in reverse, $n_{k'_1} \leqslant n_{k_1}$ and so $n_{k_1} = n_{k'_1}$ $\forall k_1, k'_1 \in \{2, ..., n\}$, $k_1 \neq k'_1$, $k_1, k'_1 \neq i$ i.e.

n_k =a constant $\forall k \in \{2, ..., n\}$ where $k \neq i$. □

Proof of Corollary to Proposition 4. Each indirect $\hat{a}_{i1(\tau)}$ in the product $\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)}$ of length 2 is of the form $\hat{a}_{i1(\tau)} = a_{ik}a_{k1}$ for some $k \in \{2, ..., n\}$, $k \neq i$, and corresponds to a path of length 2 (i.e. $i \to k \to 1$) within the spanning tree τ. And so the corollary follows immediately from Proposition 4. □

Proof of Proposition 5. Let $\tau \times \tilde{\tau}$ denote a pair of spanning trees connecting node i to node 1 by a path of length > 2 where the pair of trees are identical apart from the path connecting node i to node 1 in one tree being the reverse of the path connecting node i to node 1 in the other tree.

Let $\mathcal{T}_{k_1k_s} = \{\tau \times \tilde{\tau}$ where node i is connected to node 1 by the path $i \to k_1 \to k_2 \to ... \to k_s \to 1$ in one tree (say τ) but by the path $i \to k_s \to k_{s-1} \to ... \to k_1 \to 1$ in the other tree (say $\tilde{\tau}$), for some $\{k_1, k_s\} \subseteq \{2, ..., n\}$, $k_1, k_s \neq i$, $k_1 \neq k_s$.$\}$

$n_{k_1k_s}$ = number of pairs $\tau \times \tilde{\tau}$ in $\mathcal{T}_{k_1k_s}$.

Then if $\{k_1, k_s\} \subseteq \{2, ..., n\}$, $k_1, k_s \neq i$, $k_1 \neq k_s$ and if $\{k'_1, k'_s\} \subseteq \{2, ..., n\}$, $k'_1, k'_s \neq i$, $k'_1 \neq k'_s$ and $\{k_1, k_s\} \neq \{k'_1, k'_s\}$.

Then if $\tau \times \tilde{\tau} \in \mathcal{T}_{k_1k_s}$, then by reversing the roles of $k_1 \& k_s$ and $k'_1 \& k'_s$ respectively in $\tau \times \tilde{\tau}$, we can create another spanning tree pair $\tau' \times \tilde{\tau}' \in \mathcal{T}_{k'_1k'_s}$.

That is, if $\{k_1, k_s\} \subseteq \{2, ..., n\} k_1, k_s \neq i k_1 \neq k_s$ and if $\{k'_1, k'_s\} \subseteq \{2, ..., n\} k'_1, k'_s \neq i k'_1 \neq k'_s$ and $\{k_1, k_s\} \neq \{k'_1, k'_s\}$ and if $\tau \times \tilde{\tau} \in \mathcal{T}_{k_1k_s} \exists \tau' \times \tilde{\tau}' \in \mathcal{T}_{k'_1k'_s}$ so that $n_{k_1k_s} \leqslant n_{k'_1k'_s}$.

But using the same argument in reverse, we have $n_{k'_1k'_s} \leqslant n_{k_1k_s}$, and so

$$n_{k_1k_s} = n_{k'_1k'_s}$$
$$\forall \{k_1, k_s\} \subseteq \{2, ..., n\}, k_1, k_s \neq i, k_1 \neq k_s$$
$$\{k'_1, k'_s\} \subseteq \{2, ..., n\}, k'_1, k'_s \neq i, k'_1 \neq k'_s$$
$$\{k_1, k_s\} \neq \{k'_1, k'_s\}$$

i.e. $n_{k_1k_s}$= a constant $\forall \{k_1, k_s\} \subseteq \{2, ..., n\} k_1, k_s \neq i k_1 \neq k_s$. □

Proof of Corollary to Propotion 5.. Each indirect $\hat{a}_{i1(\tau)}$ in the product $\prod_{\tau=1}^{(n-2)n^{n-3}} \hat{a}_{i1(\tau)}$ of length $>$ 2 can be paired with a partner $\hat{a}_{i1(\tilde{\tau})}$ so that $\hat{a}_{i1(\tau)} \times \hat{a}_{i1(\tilde{\tau})}$ reduces to a product of the form $(a_{ik_1}a_{k_1 1}) \times (a_{ik_s}a_{k_s 1})$ for some $\{k_1, k_s\} \subseteq \{2, ..., n\} k_1, k_s \neq ik_1 \neq k_s$, which corresponds to a pair of paths of length $>$ 2 in a pair of spanning trees $\tau' \times \tilde{\tau}'$, and so the corollary follows immediately from Proposition 5. □

References

Aguaron, J., & Moreno-Jimenez, J. M. (2003). The geometric consistency index: approximated thresholds. *European Journal of Operational Research, 147*(1), 137–145.

Bana e Costa, C. a., & Vansnick, J.-C. (2008). A critical analysis of the eigenvalue method used to derive priorities in AHP. *European Journal of Operational Research, 187*(3), 1422–1428.

Barzilai, J. (1997). Deriving weights from pairwise comparison matrices. *Journal of the Operational Research Society, 48*(12), 1226–1232.

Barzilai, J., Cook, W. D., & Golany, B. (1987). Consistent weights for judgments matrices of the relative importance of alternatives. *Operations Research Letters, 6*(3), 131–134.

Brunelli, M., Canal, L., & Fedrizzi, M. (2013). Inconsistency indices for pairwise comparison matrices: a numerical study. *Annals of Operations Research, 211*(1), 493–509.

Brunelli, M., & Fedrizzi, M. (2015). Axiomatic properties of inconsistency indices. *Journal of the Operational Research Society, 66*(1), 1–15. doi:10.1057/jors.2013.135.

Choo, E. U., & Wedley, W. C. (2004). A common framework for deriving preference values from pairwise comparison matrices. *Computers & Operations Research, 31*(6), 893–908.

Chu, A. T. W., Kalaba, R. E., & Spingarn, K. (1979). A comparison of two methods for determining the weights of belonging to fuzzy sets. *Journal of Optimization Theory and Applications, 27*(4), 531–538.

Cook, W. D., & Kress, M. (1988). Theory and Methodology Deriving weights from pairwise comparison ratio matrices: an axiomatic approach. *European Journal Of Operational Research, 37*(3), 355–362.

Crawford, G. (1987). The geometric mean procedure for estimating the scale of a judgement matrix. *Mathematical Modelling, 9*(3–5), 327–334.

Crawford, G., & Williams, C. (1985). A note on the analysis of subjective judgment matrices. *Journal of Mathematical Psychology, 29*(4), 387–405.

Ergu, D., Kou, G., Peng, Y., Shi, Y., & Shi, Y. (2011). BIMM: a Bias Induced Matrix Model for Incomplete Reciprocal Pairwise Comparison Matrix. *Journal of Multi--Criteria Decision Analysis, 18*(1–2), 101–113.

Fedrizzi, M., & Giove, S. (2007). Incomplete pairwise comparison and consistency optimization. *European Journal of Operational Research, 183*(1), 303–313.

Fedrizzi, M., & Giove, S. (2013). Optimal sequencing in incomplete pairwise comparisons for large-dimensional problems. *International Journal of General Systems, 42*(4), 366–375.

Golany, B., & Kress, M. (1993). A multi-criteria evaluation of methods for obtaining weights from ratio-scale matrices. *European Journal of Operational Research, 69*(2), 210–220.

Harker, P. T. (1987a). Alternative modes of questioning in the analytic hierarchy process. *Mathematical Modelling, 9*(11), 353–360.

Harker, P. T. (1987b). Incomplete pairwise comparisons in the analytic hierarchy process. *Mathematical Modelling, 9*(11), 837–848.

Hartvigsen, D. (2005). Representing the strengths and directions of pairwise comparisons. *European Journal of Operational Research, 163*(2), 357–369.

Johnson, C. R. (1979). Right-left asymmetry in an eigenvector ranking procedure. *Journal of Mathematical Psychology, 19*, 61–64.

Kou, G., & Lin, C. (2014). A cosine maximization method for the priority vector derivation in ahp. *European Journal of Operational Research, 235*(2014), 225–232.

Lin, C.-C. (2007). A revised framework for deriving preference values from pairwise comparison matrices. *European Journal of Operational Research, 176*(2), 1145–1150.

Mikhailov, L. (2000). A fuzzy programming method for deriving priorities in the analytic hierarchy process. *Journal of the Operational Research Society, 51*(3), 341–349.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review, 63*(2), 81–97. http://dx.doi.org/10.1037/h0043158.

Saaty, T., & Rogers, P. (1976). Higher education in the united states (1985–200): scenario construction using hierarchical framework with eigenvector weighting. *Socio-Economic Planning Sciences, 10*, 251–263.

Saaty, T. L. (1977). A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology, 15*(3), 234–281.

Schubert, J. (2014). Partial ranking by incomplete pairwise comparisons using preference subsets. In F. Cuzzolin (Ed.), *Belief functions: Theory and applications*, Lecture Notes in Computer Science*: vol. 8764* (pp. 190–198). Springer International Publishing.

Siraj, S., Mikhailov, L., & Keane, J. A. (2012a). Enumerating all spanning trees for pairwise comparisons. *Computers & Operations Research, 39*(2), 191–199.

Siraj, S., Mikhailov, L., & Keane, J. A. (2012b). Preference elicitation from inconsistent judgments using multi-objective optimization. *European Journal of Operational Research, 220*(2), 461–471.

Siraj, S., Mikhailov, L., & Keane, J. A. (2015). Contribution of individual judgments toward inconsistency in pairwise comparisons. *European Journal of Operational Research, 242*(2), 557–567. http://dx.doi.org/10.1016/j.ejor.2014.10.024.

Srdjevic, B., Srdjevic, Z., & Blagojevic, B. (2014). First-level transitivity rule method for filling in incomplete pair-wise comparison matrices in the analytic hierarchy process. *Applied Mathematics and Information Science, 467*(2), 459–467.

Tomashevskii, I. (2015). Eigenvector ranking method as a measuring tool: Formulas for errors. *European Journal of Operational Research, 240*(3), 774–780.

Tsyganok, V. (2010). Investigation of the aggregation effectiveness of expert estimates obtained by the pairwise comparison method. *Mathematical and Computer Modelling, 52*(3–4), 538–544.

Williams, C., & Crawford, G. (1980). Analysis of Subjective Judgment Matrices. *Technical Report ADA087851*. RAND Corporation.