Int. J. Math. And Appl., 7(2)(2019), 137–148
ISSN: 2347-1557
Available Online: http://ijmaa.in/

International Journal of *Mathematics And its Applications*

Gradient Descent Method for Perron Eigenvalue Minimization of Incomplete Pairwise Comparison Matrices

Hailemariam Abebe Tekile[1,*]

1 Department of Mathematics, College of Natural and Computational Science, University of Gondar, Gondar, Ethiopia.

Abstract: Pairwise comparison matrices are often employed as a basic tool in multi-criteria decision-making. Incomplete matrix appears when some elements are missing. The paper aims to implement Gradient descent method for minimization of Perron eigenvalue problem that arises from incomplete pairwise comparison matrices. Numerical example is given at the end.

Keywords: Perron eigenvalue, Incomplete pairwise comparison matrix, Gradient descent method, Convex programming.

1. Introduction

In multi-criteria decision-making, *pairwise comparison matrices* are often used as an important key since Thomas L. Saaty proposed the Analytic Hierarchy Process (AHP) [31, 32]. A decision maker may not know the precise weights of criteria or values of the alternatives even if the comparison is possible. A pairwise comparison matrix is introduced when a decision maker wants to compare n criteria or alternatives with respect to a given criteria. It is given as $\mathbf{A} = [\alpha_{ij}]_{i,j=1,...,n}$, where α_{ij} is the numerical answer for the question 'How many times is Criterion i better than Criterion j?', or 'How many times is Alternative i preferred to Alternative j?'. It is always positive and reciprocal. Mathematically, a real matrix $\mathbf{A} = [\alpha_{ij}]_{(n\times n)}$ is called *pairwise comparison matrix* if $\alpha_{ij} > 0$ and $\alpha_{ij} = \frac{1}{\alpha_{ji}}$ for all $i, j = 1, ..., n$. A pairwise comparison matrix $\mathbf{A} = [\alpha_{ij}]_{(n\times n)}$ is called *consistent* if the transitivity $\alpha_{ij}\alpha_{jk} = \alpha_{ik}$ holds for all $i, j, k = 1, 2, ..., n$. Otherwise, it is called *inconsistent*. Pairwise comparison matrix with one or more missing elements is called *Incomplete Pairwise Comparison Matrix (IPCM)*. Missing elements of pairwise comparison matrices are introduced by variables x_i, and vector $\mathbf{x} = (x_1, x_2, ..., x_k) \in \mathbb{R}^k_+$ with structure:

$$\mathbf{A}(\mathbf{x}) = \begin{pmatrix} 1 & \alpha_{12} & x_1 & \cdots & \alpha_{1n} \\ \frac{1}{\alpha_{12}} & 1 & \alpha_{23} & \cdots & x_k \\ \frac{1}{x_1} & \frac{1}{\alpha_{23}} & 1 & \cdots & \alpha_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{1n}} & \frac{1}{x_k} & \frac{1}{\alpha_{3n}} & \cdots & 1 \end{pmatrix}. \tag{1}$$

* *E-mail: hailebb@gmail.com*

There are totally $2k$ number of variables appear in $\mathbf{A}$ (k variables in the upper triangular part and another k number of variables in its lower triangular part). After the completion of the missing elements, matrix $\mathbf{A}$ will be manipulated to verify its consistency.

The aim is to determine an appropriate positive weight vector that describes the decision maker's preference in the incomplete case. There are a number of weight estimation methods for the pairwise comparison matrix filled in by the decision maker [2, 9, 11, 19]. However, none of them are totally superior over the others. Each works best on its own standard of effectiveness. Paper [26] addressed two methods: Eigenvector Method (EM) and Logarithmic Least Squares Method (LLSM). The Eigenvector Method yields the real weight vector $\mathbf{w}^{EM}$ (right eigenvector) of a pairwise comparison matrix $\mathbf{A}$ such that

$$\mathbf{A}\mathbf{w}^{EM} = \lambda_{max}\mathbf{w}^{EM}, \tag{2}$$

where λ_{max} denotes the maximal eigenvalue, also known as Perron eigenvalue, of $\mathbf{A}$. $\mathbf{w}^{EM}$ is positive and unique up to a scalar multiplication by Perron-Frobenius Theorem. A usual normalization is given by $\sum_{i=1}^{n} w_i^{EM} = 1$.

The Logarithmic Least Squares Method yields a positive real weight vector $\mathbf{w}^{LLSM}$ as the optimal solution of the following optimization problem:

$$\min_{(w_1,\ldots,w_n)} \sum_{i=1}^{n}\sum_{j=1}^{n}\left[\log \alpha_{ij} - \log \frac{w_i}{w_j}\right]^2 \tag{3a}$$

$$\sum_{i=1}^{n} w_i = 1, w_i > 0, i = 1, 2, ..., n. \tag{3b}$$

Problem (3a)–(3b) can be solved by the geometric mean method. It is supposed that the solutions are geometrically normalized and constitutes a unique optimal solution [19].

Definition 1.1 (Inconsistency Ratio)**.** *The inconsistency ratio* CR*, according to Saaty* [31, 32]*, is defined as*

$$CR = \frac{CI}{RI_n}$$

where $CI = (\lambda_{max} - n)/(n-1)$ *and* $RI_n = \frac{\overline{\lambda}_{max} - n}{n-1}$*, and* $\overline{\lambda}_{max}$ *is an average value of the Perron eigenvalues of randomly generated* $n \times n$ *pairwise comparison matrices. Some estimated values of* RI_n *are provided in* [9, 19, 31].

It is shown that $\lambda_{max} \geq n$, and equals to n if and only if the matrix is consistent [31]. In practice one should accept matrices with values $CR \leq 0.1$ and reject values with $CR > 0.1$ [31, 36]. Note that in solving real decision problems, the weight vector $\mathbf{w}$ can be approximated by using the inconsistent pairwise comparison matrix although the levels of inconsistency are different. The real decision maker accepts some of them [7]. Different consistency indices have been presented in several literature. Saaty [31, 32] proposed the *Consistency Index* as $CI = (\lambda_{max} - n)/(n-1)$. However, the Consistency Index CI by itself is not guaranteed to compare matrices of different orders due to the variations of expectations of CI values [9]. So, it needs to be re-scaled. It must be noted that logical name of CI would be *inconsistency index*, but customarily it is called *consistency index*.

Example 1.2. *Take a pairwise comparison matrix* $\mathbf{A}$ *as*

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 8 & 1 \\ 1/3 & 1 & 1/2 & 1/7 \\ 1/8 & 2 & 1 & 3 \\ 1 & 7 & 1/3 & 1 \end{pmatrix}.$$

Here, the maximum eigenvalue, $\lambda_{max}(\mathbf{A}) = 5.3612$. *Applying the formula for CR, we have*

$$CR = \frac{CI}{RI_4} = \frac{(5.3612 - 4)/3}{0.8816} \approx 0.5146,$$

which is greater than 0.1 significantly. Hence, in Saaty sense, the matrix $\mathbf{A}$ *is too inconsistent. So, the decision has to be revised by a decision maker until* $CR < 0.1$ [19].

It is a natural phenomenon to demonstrate pairwise comparison matrices in multi-criteria decision-making process with the corresponding associated graph structures. An indirect relation through its associated undirected graph may describe two criteria that have no direct relation. For a given $n \times n$ (in)complete pairwise comparison matrix $\mathbf{A}$, its associated *undirected graph* G is given by $G := (V, E)$, where $V = \{1, 2, ..., n\}$ represents the vertices (vertices) corresponding to matrix having order n, and $E = \{e(i, j) | \alpha_{ij}$ (and α_{ji}) is given, $i \neq j\}$ denotes the undirected edges corresponding to the matrix entries. The edge is assigned exactly from vertex i to j if the comparison entry α_{ij} is already known. Missing elements in the matrix can not be described by edges.

According to Saaty's inconsistency ratio CR [31], Shiraishi, Obata and Daigo [33, 34] considered the Perron eigenvalue minimization problem for a given incomplete pairwise comparison matrix $\mathbf{A}$ as

$$\min_{\mathbf{x} \in \mathbb{R}_+^k} \lambda_{max}(\mathbf{A}(\mathbf{x})), \tag{4}$$

where $\mathbf{x} = (x_1, x_2, ..., x_k) \in \mathbb{R}_+^k$ denotes the missing entries in the matrix, and $\mathbb{R}_+^k$ represents the positive orthant of the k-dimensional Euclidean space whereas $\lambda_{max}\mathbf{A}(\mathbf{x})$ is the maximum eigenvalue that can be obtained from $(\mathbf{A}(\mathbf{x}))$ in 1.

The goal is to solve the Perron eigenvalue minimization problem (1) by using different appropriate algorithms. That means, equivalently, we need to get the best completion of the given incomplete matrix such that the minimum inconsistency ratio (CR) is satisfied [31, 32].

Theorem 1.3 ([5], Theorem 2)**.** *The optimal solution of the Perron eigenvalue minimization problem (4) is unique if and only if the graph* G *corresponding to the incomplete pairwise comparison matrix is connected.*

The Perron eigenvalue optimization problem (4) is a non-convex function of its variables. However, if graph G associated to the incomplete pairwise comparison matrix is connected, then by applying the exponential parametrization $x_1 = e^{t_1}, x_2 = e^{t_2}, ..., x_k = e^{t_k}$, it can be transformed into a strictly convex optimization problem.

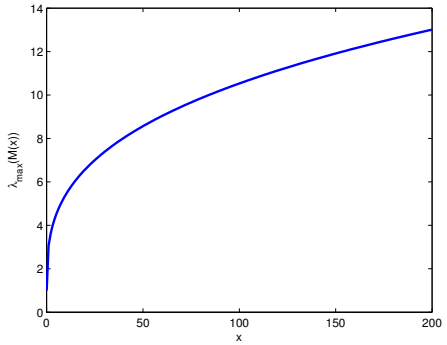

Figure 1: Non-convexity of the function $x \mapsto \lambda_{max}(\mathbf{M}(x))$ in Example 1.4

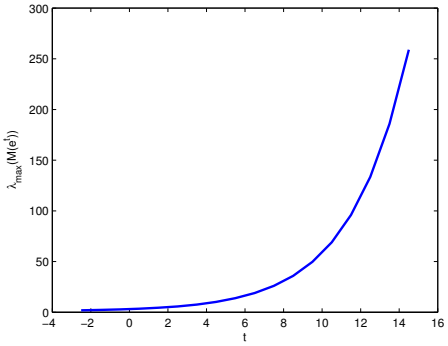

Figure 2: Strict convexity of the function $t \mapsto \lambda_{max}(\mathbf{M}(e^t))$ in Example 1.4

The formula of the maximum Perron eigenvalue λ_{max} for 3×3 incomplete pairwise comparison matrix $\mathbf{M}$ (Based on Tummala and Ling [36], and x, y, z are the variables in the upper triangular part of the matrix):

$$\mathbf{M}(x,y,z) = \begin{pmatrix} 1 & x & y \\ \frac{1}{x} & 1 & z \\ \frac{1}{y} & \frac{1}{z} & 1 \end{pmatrix}$$

is given as

$$\lambda_{max}(\mathbf{M}(x,y,z)) = 1 + \sqrt[3]{\frac{y}{xz}} + \sqrt[3]{\frac{xz}{y}}. \tag{5}$$

Example 1.4. *Let* $\mathbf{M}$ *be a* 3×3 *pairwise comparison matrix with one missing entry* x *as:*

$$\mathbf{M} = \begin{pmatrix} 1 & x & 3 \\ \frac{1}{x} & 1 & 6 \\ \frac{1}{3} & \frac{1}{6} & 1 \end{pmatrix}.$$

Then applying formula (5) and exponential parametrization $x = e^t$, we get $\lambda_{max}(\mathbf{M}(x)) = 1+\sqrt[3]{\frac{1}{2x}}+\sqrt[3]{2x}$ and $\lambda_{max}(\mathbf{M}(e^t)) = 1 + \sqrt[3]{\frac{1}{2e^t}} + \sqrt[3]{2e^t}$, respectively. Hence, it can be seen that the transformation of non-convexity of the function $\lambda_{max}(\mathbf{M}(x))$ in Figure 1 into convexity of the function $\lambda_{max}(\mathbf{M}(e^t))$ in Figure 2.

Another Logarithmic Least Squares Method for incomplete pairwise comparison matrix $\mathbf{A}$ is its extension by taking only the the given entries α_{ij}, denoted by *ILLSM*, can be defined as

$$\min_{(w_1,\ldots,w_n)} \sum_{1\le i<j\le n} \left[\log\alpha_{ij} - \log\frac{w_i}{w_j}\right]^2 + \left[\log\alpha_{ji} - \log\frac{w_j}{w_i}\right]^2 \tag{6a}$$

$$\sum_{i=1}^{n} w_i = 1, w_i > 0, i = 1, 2, ..., n. \tag{6b}$$

The objective function (6a) comprises each pair of elements related to α_{ij} and α_{ji}. The elements related to $i = j$ are 0 and omitted from the objective function.

Theorem 1.5 ([5], Theorem 4)**.** *The optimal solution of the incomplete LLSM problem (6a)–(6b) is unique if and only if graph* G *corresponding to the incomplete pairwise comparison matrix is connected.*

2. Algorithms for Perron Eigenvalue Minimization Problem

In this section, three practical Perron eigenvalue minimization algorithms are presented in order to find the best (λ_{max}-optimal) completion to the eigenvalue minimization problem (4), or equivalently, to minimize the inconsistency ratio CR. The generalization of the eigenvector method for incomplete matrix is also considered as proposed by Shiraishi, Obata and Diago [33, 34]. It is also shown that the Perron eigenvalue optimization problem can be transformed into convex optimization problem using exponential parametrization. Hence, the existence of optimal solution is guaranteed [5]. For filling in the gap of the incomplete pairwise comparison matrix as good as possible (i.e. $CR < 0.1$), one can provide an algorithm with or without derivative information. Bozóki et al. [5] proposed *cyclic coordinates* by using a general optimization function *fminbnd* in MATLAB to find the best optimal completion. Further, Ábele-Nagy [1] suggested *Newton's Method* for univariate and multivariate case to solve the Perron eigenvalue minimization problem. He used Harker's [23] first and second derivatives in his paper. In this paper, a *gradient descent method* is proposed under Section 2.3.

2.1. Cyclic Coordinate Algorithm

Cyclic coordinate's algorithm minimizes the function λ_{max} cyclically with respect to the coordinate variables iterative. It is independent of the derivatives. In the method of cyclic coordinates, the problem is divided to many, single variable minimization problems. In each iteration, a single variable function is minimized. The step is, with arbitrary or algorithmic initial value $x_i^{(0)}, i = 1, 2, ..., k$, where k represents the number of missing entries (current variables) by considering the upper triangular matrix only. Each iteration of the method consists of k steps. Thus, x_1 is changed first by fixing the rest associated variables, then x_2 and so forth through x_k. Until it achieves the stopping criteria, the process is then repeated starting with x_1 again [27].

The cyclic coordinate's algorithm is stated as follows in reference to [1, 5, 19, 27]. Here, we apply the incomplete pairwise comparison matrix $\mathbf{A}$. The goal is to find a complete matrix $\mathbf{A}$ so that λ_{max} is minimal. Let $\mathbf{x} = (x_1, x_2, ..., x_k)$, where $x_i \in \mathbf{R}_+$ are the missing entries for finite i. The value of x_i in the m^{th} step of the iteration is also supposed to be $x_i^{(m)}$.

Algorithm 1. Cyclic coordinate algorithm for $\min_{\mathbf{x}} \lambda_{max}(\mathbf{A}(\mathbf{x}))$

1: **Input:** Let $x_i^{(m)}$ denote the value of x_i in the m^{th} step of the iteration and $t_i = \log x_i \ \forall i = 1, 2, ..., k$.

2: *Set* $m \leftarrow 0$, *and* $x_i^{(0)} \leftarrow 1 \ \forall i$

3: **while** $\max_{i=1,2,..,k} ||x_i^{(m)} - x_i^{(m-1)}|| > Tolerance$ **do**

4: choose $i \in \{1, 2, .., k\}$

5: $x_i^{(m)} \leftarrow \arg\min_{x_i} \lambda_{max} A(x_1^{(m)}, ..., x_{i-1}^{(m)}, x_i, x_{i+1}^{(m-1)}, ..., x_k^{(m-1)})$

6: repeat step 5 for all $i \in \{1, 2, .., k\}$, then go to step 7

7: $m \leftarrow m + 1$

8: **end while**

end

Its global convergence is stated and proved in ([27], pages 266–267). It is difficult to compare *cyclic coordinates method* with that of the *steepest descent method* and *Newton's method*, regarding the rates of convergence, since scale factor changes do not affect it. However, rotation of coordinates affects it. Nonetheless, some comparison is possible.

2.2. Newton's Method

Newton's method is the popular method that is used to determine an optimal solution using derivative information for both univariate and multivariate cases [1]. In this section, multivariate Newton's algorithm is presented for solving Perron eigenvalue minimization problem 4. Ábele-Nagy [1] proposed the univariate and multivariate Newton methods for his optimal completion by following Bozóki et al. [5] paper. In the univariate case, in order to optimize only one variable at a time, he used the method of cyclic coordinates with Newton iteration. By using Newton's Multivariate method, it is possible to optimize all the variables at the same time rather than optimizing one variable at a time.

The steps for multivariate Newton's method is given as follows in reference to [1]. Let $x = e^t$ be in the position of (i, j) in the incomplete matrix and $R(t) = \lambda_{max}(e^{t_1}, e^{t_2}, ..., e^{t_k})$, where k is the number of elements in the upper triangular matrix at a time. The derivatives $\frac{\partial \lambda_{max}(x)}{\partial x}$ and $\frac{\partial^2 \lambda_{max}(x)}{(\partial x)^2}$ are known by Harker [20]. Here, the aim is to minimize R. To do so, Newton's iteration is used as

$$t^{(m+1)} = t^{(m)} - \gamma[HR(t^{(m)})]^{-1} \nabla R((t^{(m)}), \tag{7}$$

where $HR(t^{(m)})$ denotes the Hessian matrix of $R(t)$, $\nabla R(t^{(m)})$ is the gradient vector of $R(t)$, and the step size γ. The gradient

vector $\nabla R(t) = (\frac{\partial R}{\partial t_1}, ..., \frac{\partial R}{\partial t_k})$ can be found from the following formula [1, 20] and it is also appeared in the Appendix (4):

$$\frac{\partial R(t)}{\partial t} = \frac{\partial \lambda_{max}(e^t)}{\partial t} = \frac{\partial \lambda_{max}(x)}{\partial x} \cdot \frac{\partial (e^t)}{\partial t} = \frac{\partial \lambda_{max}(x)}{\partial x} \cdot e^t. \tag{8}$$

Similarly, from the paper ([1], page 65), the derivation for the Hessian matrix has been given. The stopping criteria must be given for x, but not for t, as small changes in t results in large differences in x.

Algorithm 2. Newton's Multivariate Algorithm for $\min_{\mathbf{x}} \lambda_{max}(\mathbf{A(x)})$

1: **Input:** Let $t^{(m)}$ denote the value of t in the m^{th} iteration, and $x_i^{(m)}$ be the value of x_i in the m^{th} iteration and $t_i = \log x_i$ $\forall i = 1, 2, ..., k$.

2: *Starting values of* t, where $t = \log x$, and *step size* γ

3: *Set* $m \leftarrow 0$

4: **while** $\max_{i=1,2,..,k} ||x_i^{(m)} - x_i^{(m-1)}|| > Tolerance$ **do**

5: $t^{(m+1)} \leftarrow t^{(m)} - \gamma[HR(t^{(m)})]^{-1} \nabla R(t^{(m)})$

6: $m \leftarrow m + 1$.

7: **end while**

end

Newton's method will have a quadratic convergence by assuming the Jacobian matrix at the current point is has no inverse. Otherwise, the global converge may not be guaranteed. Its global convergence is stated and proved in Numerical Optimization, Nocedal and Wright's book ([30], pages 44–45).

2.3. Gradient Descent Method

Gradient descent method is a first-order derivative iterative optimization method by considering steps proportional to the negative of the gradient function at the current point. In this section, the *gradient descent method* is implemented for solving the eigenvalue minimization problem (4). Let $R(t) = \lambda_{max}(e^{t_1}, e^{t_2}, ..., e^{t_k})$ with parametrization $x = e^t$, where k denotes the number of missing elements in the upper triangular incomplete matrix 1. The goal is to minimize $R(t)$. The gradient function $\nabla R(t) = (\frac{\partial R}{\partial t_1}, ..., \frac{\partial R}{\partial t_k})$ is derived from $R(t)$ in equation (**??**). (See also the appendix for Harker's [20] first derivative formula). For a given point $t = \bar{t}$, the approximation of $R(t)$ can be presented as a linear approximation because $R(t)$ is convex and differentiable with small norm $||d||$:

$$R(\bar{t} + d) \approx R(\bar{t}) + \nabla R(\bar{t})^T d$$

Now we choose d so as to make the scalar product $\nabla R(\bar{t})^T d$ is as small as possible. A particular direction $d^* = \frac{-\nabla R(\bar{t})}{||\nabla R(\bar{t})||}$ provides the smallest scalar product with the gradient $\nabla R(\bar{t})$ by normalizing d with $||d|| = 1$. This reality can be asserted by the following inequality [16, 19]:

$$\nabla R(\bar{t})^T d \geq -||\nabla R(\bar{t})|| \cdot ||d|| = \nabla R(\bar{t})^T \left(\frac{-\nabla R(\bar{t})}{||\nabla R(\bar{t})||} \right) = \nabla R(\bar{t})^T d^*.$$

As a consequence, at the current point $\bar{t}$, the denormalized direction $\bar{d} = -\nabla R(\bar{t})$ is called the *direction of gradient descent*, or simply *descent direction*. It is important to note that $\bar{d}^T \nabla R(\bar{t}) = -\nabla R(\bar{t})^T \nabla R(\bar{t}) < 0$ only if $\nabla R(\bar{t}) \neq 0$, i.e. $\bar{d}$ is the descent direction provided that $\nabla R(\bar{t}) \neq 0$. As small changes in t results in large differences in x, the stopping criteria must be given for x but not for t. In each iteration, the locally optimal step size γ maybe chosen by an exact or inexact *line*

search algorithm or just by fixed γ. However, doing *line search* can be time-taking. Using a fixed γ on the other hand can result in poor convergence. To get fewer iterations, it can be better options by using Newton's method and Hessian versions of conjugate gradient methods. But, each iteration cost is higher. Nonetheless, the gradient descent algorithm works well in any number of dimension. Its global convergence is stated and proved in ([16], pages 3–5). The algorithm is constructed as follows.

Algorithm 3. Gradient Descent Algorithm for $\min_{\mathbf{x}} \lambda_{max}(\mathbf{A}(\mathbf{x}))$

1: **Input:** Let t^m denote the value of t in the m^{th} iteration, and x_i^m be the value of x_i in the m^{th} iteration where $t_i = \log x_i$ $\forall i = 1, 2, ..., k$.
2: *Choose starting values of* t *and step size* γ
3: *Set* $m \leftarrow 0$
4: **while** $\max_{i=1,2,..,k} ||x_i^{(m)} - x_i^{(m-1)}|| > Tolerance$ **do**
5: $d^m \leftarrow -\nabla R(t^m)$
6: *If* $d^m = 0$, *then* stop. Otherwise, go to step 7
7: $t^{m+1} \leftarrow t^m + \gamma d^m$
8: $m \leftarrow m + 1$
9: **end while**

end

3. Numerical Illustration

Consider an 8×8 incomplete pairwise comparison matrix (Saaty's 'buying a house' incomplete version [31]) as follows:

$$\mathbf{A}(\mathbf{x}) = \begin{pmatrix} 1 & 5 & 3 & 7 & 6 & 6 & 1/3 & 1/4 \\ 1/5 & 1 & x_1 & 5 & x_2 & 3 & x_3 & 1/7 \\ 1/3 & 1/x_1 & 1 & x_4 & 3 & x_5 & 6 & x_6 \\ 1/7 & 1/5 & 1/x_4 & 1 & x_7 & 1/4 & x_8 & 1/8 \\ 1/6 & 1/x_2 & 1/3 & 1/x_7 & 1 & x_9 & 1/5 & x_{10} \\ 1/6 & 1/3 & 1/x_5 & 4 & 1/x_9 & 1 & x_{11} & 1/6 \\ 3 & 1/x_3 & 1/6 & 1/x_8 & 5 & 1/x_{11} & 1 & x_{12} \\ 4 & 7 & 1/x_6 & 8 & 1/x_{10} & 6 & 1/x_{12} & 1 \end{pmatrix}.$$

Both Bozóki et al. [5] and Ábele-Nagy [1] have used this matrix as an example in their papers. It is worth noting that the graph associated to the matrix is connected [5]: the first row is completely filled in, and node 1 is directly connected to all nodes.

The aim is to get a complete matrix $\mathbf{A}$ so that λ_{max} is minimal by applying gradient descent algorithm. In this matrix, the missing entries are represented by vector $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}) \in \mathbb{R}_+^{12}$. The approximated values (up to 4 digits) for all variables in each iteration are given in Table 1 based on the algorithm. Moreover, the tolerance $T = 10^{-4}$ and initial value $x_k^{(0)}$ have been used by taking the weights (values) from Incomplete Logarithmic Least Squares Method (ILLSM) in [1, 8, 19] with regard to an appropriate position of $x_k^{(0)}$ in the matrix (i, j) such that $x_k^{(0)} = w_i/w_j$ for $k = 1, ..., 12$. After several numerical experiments, the value for step size $\gamma = 6.77$ is applied in order to get the minimum

number of iterations. Hence, 21 iterations (the least possible number of iterations) were obtained which can be seen in Table 1. The gradient descent algorithm gives 25 iterations with $\gamma = 6.55$ when the starting value $x_k^{(0)} = 1 \, \forall k = 1, ..., 12$. But, its number of iterations becomes higher(for instance, 37 with $\gamma = 6.77$). Newton's method with ILLSM for $x_k^{(0)}$ and $\gamma = 0.45$ yield $m = 14$ number of iterations in both univariate and multivariate case. But, with starting value $x_k^{(0)} = 1$, again $\gamma = 0.45$ resulted in $m = 15$ in the univariate case while $m = 26$ iterations in the multivariate case [1]. However, Newton's multivariate method has provided the least possible iterations ($m = 20$) with $\gamma = 1.77$ and $x_k^{(0)} = 1$ for all $k = 1, 2, .., 12$. In addition to this, it can be seen that in Figure 3 the objective function's value decreases in the first few iterations (with 4 decimal place approximation). However, if we take 15 decimal places, $\lambda_{max}(\mathbf{A}(\mathbf{x}))$ values decrease throughout the iterations. For instance, λ_{max} value for the 20th and 21st iterations is 9.298092250616087 and 9.298092250477843, respectively [19].

m	$x_1^{(m)}$	$x_2^{(m)}$	$x_3^{(m)}$	$x_4^{(m)}$	$x_5^{(m)}$	$x_6^{(m)}$	$x_7^{(m)}$	$x_8^{(m)}$	$x_9^{(m)}$	$x_{10}^{(m)}$	$x_{11}^{(m)}$	$x_{12}^{(m)}$
0	0.3823	1.8428	0.4758	8.9924	4.2688	0.5228	0.5361	0.1384	0.8855	0.1085	0.2916	0.4200
1	0.3338	1.7890	0.4866	9.7728	4.7893	0.5609	0.5426	0.1495	0.8975	0.1061	0.3045	0.3834
2	0.3384	1.7603	0.4746	9.6902	4.7370	0.5565	0.5396	0.1445	0.9084	0.1063	0.2961	0.3979
3	0.3341	1.7529	0.4727	9.7944	4.7905	0.5626	0.5344	0.1446	0.9142	0.1075	0.2953	0.3967
4	0.3333	1.7403	0.4710	9.8250	4.8064	0.5641	0.5320	0.1437	0.9199	0.1079	0.2941	0.3999
5	0.3322	1.7351	0.4692	9.8566	4.8199	0.5297	0.5297	0.1434	0.9231	0.1084	0.2930	0.4002
6	0.3315	1.7298	0.4686	9.8752	4.8313	0.5669	0.5285	0.1431	0.9257	0.1086	0.2926	0.4015
7	0.3311	1.7270	0.4677	9.8899	4.8365	0.5680	0.5274	0.1429	0.9273	0.1088	0.2921	0.4017
8	0.3307	1.7246	0.4674	9.8987	4.8425	0.5683	0.5267	0.1427	0.9285	0.1090	0.2919	0.2918
9	0.3305	1.7232	0.4670	9.9059	4.8447	0.5688	0.5263	0.1427	0.9293	0.1091	0.2916	0.4024
10	0.3303	1.7221	0.4669	9.9100	4.8478	0.5690	0.5260	0.1426	0.9299	0.1092	0.2915	0.4027
11	0.3303	1.7214	0.4667	9.9136	4.8487	0.5693	0.5257	0.1425	0.9303	0.1092	0.2914	0.4028
12	0.3302	1.7209	0.4666	9.9155	4.8503	0.5693	0.5256	0.1425	0.9306	0.1092	0.2914	0.4029
13	0.3301	1.7205	0.4665	9.9173	4.8507	0.5694	0.5255	0.1425	0.9308	0.1092	0.2913	0.4029
14	0.3301	1.7203	0.4665	9.9182	4.8515	0.5695	0.5254	0.1425	0.9309	0.1093	0.2913	0.4030
15	0.3301	1.7201	0.4664	9.9190	4.8516	0.5695	0.5254	0.1425	0.9310	0.1093	0.2913	0.4030
16	0.3300	1.7199	0.4664	9.9194	4.8520	0.5695	0.5253	0.1424	0.9310	0.1093	0.2912	0.4030
17	0.3300	1.7199	0.4663	9.9198	4.8520	0.5695	0.5253	0.1424	0.9311	0.1093	0.2912	0.4030
18	0.3300	1.7198	0.4663	9.9200	4.8522	0.5695	0.5253	0.1424	0.9311	0.1093	0.2912	0.4030
19	0.3300	1.7198	0.4663	9.9202	4.8523	0.5695	0.5252	0.1424	0.9311	0.1093	0.2912	0.4030
20	0.3300	1.7198	0.4663	9.9204	4.8524	0.5695	0.5253	0.1424	0.9312	0.1093	0.2912	0.4030
21	0.3300	1.7198	0.4663	9.9205	4.8524	0.5696	0.5253	0.1424	0.9312	0.1093	0.2912	0.4031

Table 1: The 21 iterations of the gradient descent algorithm to the given incomplete pairwise comparison matrix $\mathbf{A}$.

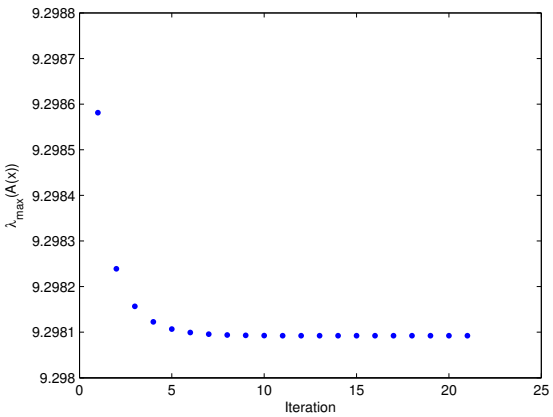

Figure 3: $\lambda_{max}(\mathbf{A}(\mathbf{x}))$ values for the first 21 iterations

Based on the algorithm results from Table 1, the optimal solution of the Perron eigenvalue minimization problem for the given incomplete matrix $\mathbf{A}$ is presented as follows. Here, the missing entries are given with four digits accuracy. The algorithm results in the same value with that of Bozóki et al. [5] and Ábele-Nagy [1], except few variants in the third and fourth decimal places (missing values are taken from the last iteration). Hence, we've the completed matrix as

$$\mathbf{A}(\mathbf{x}^*) = \begin{pmatrix} 1 & 5 & 3 & 7 & 6 & 6 & 1/3 & 1/4 \\ 1/5 & 1 & \mathbf{0.3300} & 5 & \mathbf{1.7198} & 3 & \mathbf{0.4664} & 1/7 \\ 1/3 & \mathbf{3.0303} & 1 & \mathbf{9.9205} & 3 & \mathbf{4.8524} & 6 & \mathbf{0.5696} \\ 1/7 & 1/5 & \mathbf{0.1008} & 1 & \mathbf{0.5253} & 1/4 & \mathbf{0.1424} & 1/8 \\ 1/6 & \mathbf{0.5815} & 1/3 & \mathbf{1.9040} & 1 & \mathbf{0.9312} & 1/5 & \mathbf{0.1093} \\ 1/6 & 1/3 & \mathbf{0.2061} & 4 & \mathbf{1.0740} & 1 & \mathbf{0.2912} & 1/6 \\ 3 & \mathbf{2.1445} & 1/6 & \mathbf{7.0225} & 5 & \mathbf{3.4341} & 1 & \mathbf{0.4030} \\ 4 & 7 & \mathbf{1.7556} & 8 & \mathbf{9.1491} & 6 & \mathbf{2.4814} & 1 \end{pmatrix}.$$

The minimum value of the objective function $\lambda_{max}(\mathbf{A}(\mathbf{x}^*))$ is 9.2981. The associated normalized right eigenvector is

$$\mathbf{w}^{EM} = (0.1894, 0.0567, 0.2116, 0.0175, 0.0319, 0.0354, 0.1509, 0.3066)^T.$$

Its inconsistency ratio can be evaluated as

$$CR = \frac{CI}{RI_8} = \frac{(9.2981 - 8)/7}{1.4057} \approx 0.1319,$$

which is above the threshold (i.e. $CR > 0.1$). Hence, the matrix is a bit above acceptable inconsistency according to Saaty's criteria [19, 31, 32].

4. Conclusion

The necessary and sufficient conditions for the optimal completion of a given incomplete pairwise comparison matrix is the connectedness of an associated graph 1.3-1.5. In this paper, implementation of Gradient descent method for Perron eigenvalue minimization of incomplete pairwise comparison matrices is presented. Since the Perron eigenvalue minimization problem can be transformed into convex problem (strictly convex in the case of connected graph), the global convergence of the algorithms (cyclic coordinates method, Newton's method, and gradient descent method) is guaranteed. However, the choice of step size still affects the stability of the algorithms. The future research can be the choice of step size γ and comparison analysis of the algorithms.

Acknowledgement

The author wishes to thank Mr. George Soros for his financial support at Central European University while he was doing his research. He would also like to thank Prof. Sandor Bozoki for his help.

References

[1] K. Ábele-Nagy, *Minimization of Perron eigenvalue of incomplete pairwise comparison matrices by Newton iteration*, Acta Univ. Sapientiae, Informatica 7(1)(2015), 58-71.

[2] G. Bajwa, E. U. Choo and W. C. Wedley, *Effectiveness Analysis of Deriving Priority Vectors from Reciprocal Pairwise Comparison Matrices*, Asia-Pacific Journal of Operational Research, 25(3)(2008), 279-299.

[3] M. S. Bazaraa, D. S. Hanif and C. M. Shetty, *Nonlinear programming: theory and algorithms*, Second Edition, John Wiley and Sons, Inc., (1993).

[4] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, (2004).

[5] S. Bozóki, J. Fülöp and L. Rónyai, *On optimal completions of incomplete pairwise comparison matrices*, Mathematical and Computer Modelling, 52(2010), 318-333.

[6] S. Bozóki, *Two short proofs regarding the logarthmic least squares optimality*, Annals of Operations Research, DOI 10.1007/s10479-016-2396-9.

[7] S. Bozóki and T.Rapcsák, *On Saaty's and Koczkodaj's inconsistencies of pairwise comparison matrices*, Journal of Global Optimization, 42(2)(2008), 157-175.

[8] R. P. Brent, *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ, (1973).

[9] M. Brunelli, *Introduction to the Analytic Hierarchy Process*, (2015).

[10] K. Chen, G. Kou, J.M. Tarn and Y. Song, *Bridging the gap between missing and inconsistent values in eliciting preference from pairwise comparison matrices*, Annals of Operations Research, 235(1)(2015), 155-175.

[11] E. U. Choo and W. C. Wedley, *A common framework for deriving preference values from pairwise comparison matrices*, Computers and Operations Research, 31(2004), 893-908.

[12] G. Crawford and C. Williams, *A note on the analysis of subjective judgment matrices*, Journal of Mathematical Psychology, 29(4)(1985), 387-405.

[13] L. Csató, *Ranking by pairwise comparisons for Swiss-system tournaments*, Central European Journal of Operations Research, 21(4)(2013), 783-803.

[14] G. Dahl *A matrix-based ranking method with application to tennis*, Linear Algebra and its Applications, 437(1)(2012), 26-36.

[15] P. De Jong, *A statistical approach to Saatys scaling methods for priorities*, Journal of Mathematical Psychology, 28(1984), 467-478.

[16] R. M. Freund, *The Steepest Descent Algorithm for Unconstrained Optimization and a Bisection Line-search Method*, Massachusetts Institute of Technology, (2004).

[17] J. Fülöp, *An Optimization Approach for the Eigenvector Method*, IEEE 8th International symposium on Applied Computational Intelligence and Informatics (SACI 2013), Timisoara, Romania, 23–25 May 2013.

[18] B. Golany and M. Kress, *A multicriteria evaluation of methods for obtaining weights from ratio-scale matrices*, European Journal of Operational Research, 69(1993), 210-220.

[19] T. Hailemariam, *A Thesis on Incomplete Pairwise Comparison Matrices in Multi-Criteria Decision Making and Ranking*, https://mathematics.ceu.edu/sites/mathematics.ceu.hu/files/attachment/basicpage/29/thesishailemariam.pdf

[20] P. T. Harker, *Derivatives of the Perron root of a positive reciprocal matrix: with application to the analytic hierarchy process*, Applied Mathematics and Computation, 22(2)(1987), 217-232.

[21] P. T. Harker, *Alternative modes of questioning in the analytic hierarchy process*, Mathematical Modelling, 9(3)(1987), 353-360.

[22] P. T. Harker, *Incomplete pairwise comparisons in the analytic hierarchy process*, Mathematical Modelling, 9(11)(1987), 837-848.

[23] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, (1985).

[24] A. Ishizaka and M. Lusti, *How to derive priorities in AHP: a comparative study*, Central European Journal of Operations Research, 14(2006), 387-400.

[25] J. F. C. Kingman, *A convexity property of positive matrices*, The Quarterly Journal of Mathematics, 12(1961), 283-284.

[26] M. Kwiesielewicz, *The logarithmic least squares and the generalised pseudoinverse in estimating ratios*, European Journal of Operational Research, 93(1996), 611-619.

[27] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, Series: International Series in Operations Research and Management Science, Vol. 116, 3rd Edition, (Springer, 2008).

[28] MATLAB documentation: *fminbnd*, http://www.mathworks.com/help/matlab/ref/fminbnd.html.

[29] MATLAB from youtube: Gradient descent, https://www.mathworks.com/matlabcentral/fileexchange/35535-simplified-gradient-descent-optimization.

[30] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd Ed., Springer Series in Operations Research, (2006).

[31] T. L. Saaty, *The analytic hierarchy process*, McGraw-Hill, New York, (1980).

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

[33] S. Shiraishi and T. Obata, *On a maximization problem arising from a positive reciprocal matrix in AHP*, Bulletin of Informatics and Cybernetics, 34(2)(2002), 91-96.

[34] S. Shiraishi, T. Obata and M. Daigo, *Properties of a positive reciprocal matrix and their application to AHP*, Journal of the Operations Research Society of Japan, 41(3)(1998), 40-414.

[35] E. Triantaphyllou, B. Shu, S. Nieto Sanchez and T. Ray, *Multi-Criteria Decision Making: An Operations Research Approach*, Encyclopedia of Electrical and Electronics Engineering, (J.G. Webster, Ed.), John Wiley and Sons, New York, NY, 15(1998), 175-186.

[36] V. M. R. Tummala and H. Ling, *A note on the computation of the mean random consistency index of the analytic hierarchy process (AHP)*, Theory and Decision, 44(1998), 221-230.

[37] O. S. Vaidya and S. Kumar, *Analytic hierarchy process: An overview of applications*, European Journal of Operational Research, 169(2006), 1-29.

[38] S. J. Wright, *Coordinate Descent Algorithms*, http://www.optimization-online.org/DB_FILE/2014/12/4679.pdf

Appendix

The following Harker's [20] formulas are associated to Section 2. Harker's formulas for the derivatives of the Perron eigenvalue are presented as follows [1, 19, 20]. Let $\mathbf{A}$ denote a pairwise comparison matrix, and let $x = x(\mathbf{A})$ denote the left Perron eigenvectors, $y = y(\mathbf{A})$ denote its right Perron eigenvector and $\lambda_{max} = \lambda_{max}(\mathbf{A})$ its Perron eigenvalue, so $\mathbf{A}x = \lambda_{max}x$ and $y^T\mathbf{A} = \lambda_{max}y^T$. The normalization for the eigenvectors in this case is $y^Tx = 1$. Let $Q = \lambda_{max}I - \mathbf{A}$. Also let Q^+ denote the pseudoinverse of Q, with properties: $QQ^+Q = Q$, $Q^+QQ^+ = Q^+$, $Q^+Q = QQ^+$. Further, $\partial\alpha_{ij}$ represents differentiation with respect to the entry in position (i, j) in $\mathbf{A}$, and similarly $\partial\alpha_{kl}$ represents differentiation with respect to the entry in position (k, l). Applying these notations, the formulas are given as follows:

- **Harkers formula for the first derivative** $\frac{\partial\lambda_{max}(x)}{\partial x}$:

$$\left(\frac{\partial\lambda_{max}}{\partial_{ij}}\|i > j\right) = \left([y_ix_j] - \frac{[y_jx_i]}{[\alpha_{ij}]^2}\right)$$

where vectors $x = x(\mathbf{A}), y = y(\mathbf{A})$ are the right-hand side and left-hand side eigenvectors of $\mathbf{A}$, respectively.

- **Harker's formula for the second derivative (when $i \neq k$ or $j \neq l$):**

$$\frac{\partial^2 \lambda_{max}}{\partial_{ij}\partial_{kl}} = (xy^T)_{li}Q^+_{jk} + (xy^T)_{jk}Q^+_{li} - \frac{(xy^T)_{ki}Q^+_{jl} + (xy^T)_{jl}Q^+_{ki}}{[\alpha_{kl}]^2} - \frac{(xy^T)_{lj}Q^+_{ik} + (xy^T)_{ik}Q^+_{lj}}{[\alpha_{ij}]^2} - \frac{(xy^T)_{kl}Q^+_{il} + (xy^T)_{il}Q^+_{kj}}{[\alpha_{ij}]^2[\alpha_{kl}]^2}.$$

- **Harker's formula for the second derivative (when $i \neq k$ or $j \neq l$):**

$$\frac{\partial^2 \lambda_{max}}{\partial_{ij}\partial_{kl}} = \frac{2(xy^T)_{ij}}{[\alpha_{ij}]^3} + 2(xy^T)_{ji}Q^+_{ii} - 2\frac{(xy^T)_{ii}Q^+_{jj} + (xy^T)_{jj}Q^+_{ii}}{[\alpha_{ij}]^2} + 2\frac{(xy^T)_{ij}Q^+_{ij}}{[\alpha_{ij}]^4}$$