FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 10 No. 6, March, 2026, pp 80 – 87
DOI: https://doi.org/10.33003/fjs-2026-1006-4702

AN AUTOMATED CONSISTENCY ADJUSTMENT ALGORITHM FOR DELPHI–ANALYTICAL HIERARCHY PROCESS MODELS: DESIGN, IMPLEMENTATION, AND VALIDATION

***[1]Muyideen Adeyemi Bello, [1]Juliana Ndunagu, [1]Ismaeel Ahmad Barroon, [1]Adekunle Musa Ajibade, [2]Ameerat Olorunkemi Bello and [1]Nuraddeen Ibrahim Isa**

[1]National Open University of Nigeria, African Centre of Excellence On Technology Enhanced Learning, Abuja, Nigeria.
[2]Department of Computing, East Tennessee State University, Johnson City, United States of America.

*Corresponding authors' email: ace22160001@noun.edu.ng, mabello63@gmail.com Phone: +2348034713459

ABSTRACT

Despite the widespread use of the Delphi and Analytic Hierarchy Process (AHP) methods for prioritizing complex problems, a persistent limitation is the persistence of inconsistencies in expert judgment matrices during the AHP data elicitation stage. Existing solutions, particularly genetic algorithm–based approaches, often entail high computational complexity due to population-based search, crossover, and mutation operations, making them less efficient and more difficult to implement in practical settings. There is therefore a need for a simpler, deterministic, and practically implementable consistency ratio (CR) adjustment mechanism within Delphi–AHP frameworks. To address this problem, this study develops a deterministic, automated consistency adjustment algorithm that efficiently corrects inconsistencies in AHP pairwise comparison matrices without relying on genetic algorithm mechanisms. The proposed algorithm was evaluated through controlled experiments on synthetic matrices with varying levels of inconsistency. A design science research methodology guided the development and implementation of the algorithm in a C# web-based decision support system. The framework was validated empirically using data collected from a Delphi–AHP study involving e-learning experts from public and private Nigerian

Keywords: Delphi-AHP hybridization, Consistency Ratio, Judgment Matrix Adjustment, Deterministic Algorithm, Decision Support Systems

INTRODUCTION

The integration of the Delphi technique and the Analytic Hierarchy Process (AHP) has become a widely adopted methodological framework for structuring complex, expert-driven decision problems across public policy, management, and technological domains (Humphrey-Murto & de Wit, 2019; Muhammad et al., 2020; Han, 2021). By combining structured consensus-building with hierarchical prioritization, Delphi–AHP models enable systematic evaluation of multifaceted challenges where empirical data may be limited or uncertain (Saaty, 1987; Saaty et al., 2012).

Despite its methodological appeal, the operational reliability of Delphi–AHP implementations depends critically on the internal coherence of expert pairwise comparison matrices. Judgment inconsistency remains one of the most frequently reported methodological challenges in applied AHP studies (Madzík & Falát, 2022; Liu et al., 2023). Within the AHP framework, consistency is typically assessed using the consistency ratio (CR), introduced by Saaty (1977, 1987), with values exceeding the commonly accepted threshold of 0.10 indicating insufficient logical coherence for reliable priority derivation.

In practice, managing inconsistency presents nontrivial challenges. In survey-based Delphi environments, experts may be geographically dispersed or constrained by limited availability, making iterative re-engagement impractical (Amenta et al., 2021). Consequently, researchers often exclude inconsistent matrices from further analysis (Farid et al., 2015; Han, 2021), potentially reducing sample representativeness. Alternatively, computational approaches based on stochastic or population-based optimization, including genetic algorithms and related metaheuristics, have been proposed to minimize inconsistency indices (Li et al., 2020; Zhang et al., 2022). While flexible, such approaches may introduce parameter sensitivity, stochastic variability, and increased computational overhead, thereby complicating reproducibility and system integration.

From a software engineering perspective, these interventions affect the stability and efficiency of decision-support workflows. Manual revision cycles increase iteration time and user dependency, whereas metaheuristic optimization methods may reduce transparency and complicate maintainability within operational decision-support systems (Pressman & Maxim, 2020; Sommerville, 2021). Recent research on decision-support system reliability emphasizes the importance of embedded quality-control mechanisms that enhance repeatability and reduce human rework cycles (Liu et al., 2023; Zhang et al., 2022). However, comparatively limited attention has been devoted to deterministic, rule-driven adjustment procedures specifically designed for seamless integration within Delphi–AHP software environments.

This study addresses this gap by proposing a deterministic algorithm for automated consistency adjustment in Delphi–AHP models. Rather than treating inconsistency reduction solely as an optimization problem, the proposed method conceptualizes adjustment as a structured reliability-enhancement mechanism embedded within the decision workflow. The algorithm iteratively modifies selected pairwise comparison entries according to the implied priority structure of the matrix, enforcing reciprocity and structural coherence without reliance on stochastic population-based operators or extensive parameter tuning.

To evaluate operational feasibility, the algorithm was implemented in a C#-based web application that the authors developed and tested using both synthetically generated matrices with controlled levels of inconsistency and empirical data from a real Delphi–AHP case study. The experimental results demonstrate a consistent reduction of CR below accepted thresholds while preserving ranking stability across practical matrix sizes.

By embedding automated consistency stabilization directly into the decision-support lifecycle, this work contributes to the evolution of quality assurance mechanisms in expert-based systems. The proposed approach enhances computational predictability, reproducibility, and workflow reliability, thereby supporting the maturation of Delphi–AHP implementations within software-enabled decision environments.

The remainder of this article is organized as follows. Section 2 reviews related literature on Delphi–AHP applications and existing inconsistency adjustment techniques. Section 3 presents the proposed algorithm and its integration within a decision-support framework. Section 4 details the experimental design and empirical evaluation. Section 5 discusses process-level implications and scalability considerations. Section 6 concludes with directions for future research

MATERIALS AND METHODS

AHP Concepts

The Analytic Hierarchy Process (AHP) applies a structured approach to problem-solving based on three foundational principles: decomposition, comparative judgment, and synthesis of priorities. Together, these principles enable complex decision problems to be analysed systematically and transparently (Sorooshian et al., 2015). The first principle, decomposition, involves structuring the components of a decision problem into a hierarchical framework. This hierarchy typically begins with an overall goal at the highest level, followed by criteria and, where necessary, sub-criteria, and finally the decision alternatives at the lowest level. Decomposition allows complex problems to be broken down into manageable elements that can be evaluated independently before being recombined into a coherent decision outcome. The second principle, comparative judgment, entails evaluating decision elements through pairwise comparisons. At each level of the hierarchy, elements are compared with respect to their relative importance toward an element at the immediately higher level. These judgments are organised into pairwise comparison matrices, from which priority weights are derived. The principal right eigenvector of each matrix is used to obtain ratio-scaled priority values that quantify the relative importance of the compared elements. The third principle, synthesis of priorities, involves aggregating the local priority weights across the hierarchy to compute the global priorities of the decision alternatives. This synthesis process ensures that the final ranking of alternatives reflects the combined influence of all criteria and sub-criteria considered in the model, thereby supporting rational and consistent decision-making.

Building on these principles, Thomas L. Saaty (1987) and subsequent refinements by Saaty et al. (2012) outlined a sequence of procedural steps that form the core of the AHP methodology. In this study, the AHP hierarchy constructed through the decomposition process consists of three main levels: the *goal*, the *criteria*, and the *alternatives*. The goal represents the overall objective of the decision problem, which in this research is the prioritization of e-learning challenges confronting higher education institutions in a developing-country context. The criteria correspond to the key factors that influence this decision and may be further decomposed into sub-criteria depending on the complexity of the problem domain. The alternatives represent the competing options or outcomes evaluated in relation to the stated goal. The structure of this hierarchical arrangement is illustrated in Figure 1.

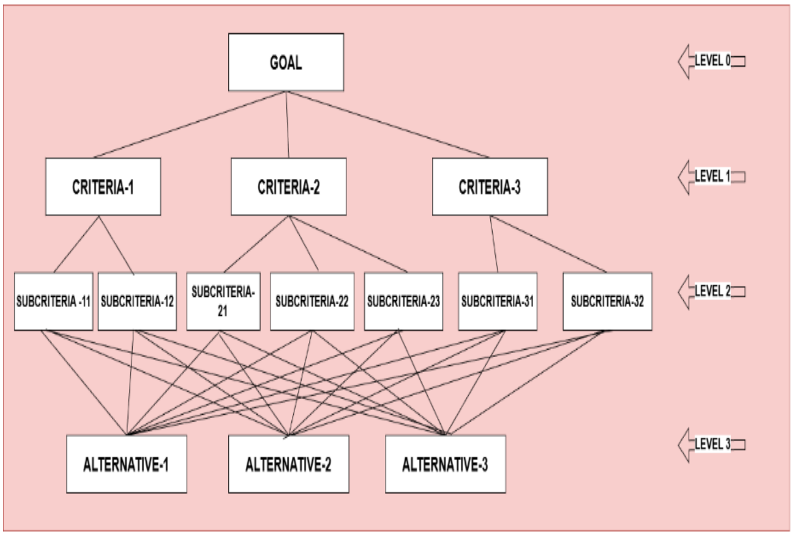

Figure 1: Construction of a Four-level AHP Decision Hierarchy

Once the hierarchy is established, comparative judgments, technically referred to as *pairwise comparisons*, are conducted at each level to determine the relative importance of decision elements. These comparisons express the degree to which one element is preferred over another and are organised into pairwise comparison matrices. To quantify these judgments, decision-makers assign numerical values using the 9-point intensity of relative importance scale, which captures varying levels of preference between paired elements, as presented in Table 1.

Mathematical Basis for AHP

Definitions

Given a square matrix of order n, mathematically represented as A:

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} \qquad (1)$$

where $a_{ij} > 0$ and for all i, j = 1,2,3, …, n.

If $a_{ij} = 1$ whenever i = j and $\frac{1}{a_{ji}} = a_{ij}$ whenever i ≠ j, then A is a positive reciprocal matrix of order n.
There are two ways we can generate A (Pant et al. 2022). Firstly, let Q be a matrix of order n with each element equal to 1. A nontrivial reciprocal matrix of the same order can be generated with the help of this matrix Q. A nontrivial reciprocal matrix is a positive reciprocal matrix whose entries are not all necessarily 1. For this nontrivial case, let us define a diagonal matrix, $D = diag(d1, d2, ..., dn)$ of order n having positive diagonal entries. D is neither an identity nor a null matrix. Then the matrix $A = DQD^{-1}$ is a positive reciprocal matrix.
A second method to generate A = $[a_{ij}]$ of order n, is by calculating $a_{ij} = w_i / w_j$ where w_i, w_j are the elements of a finite set $W = \{w_1, w_2, . . . , w_n: w_i \in R, i = 1, 2, . . . n\}$
According to Saaty (1977), if $w = \{w_1, w_2, . . . , w_n: w_i \in R, i = 1, 2, . . . n\}$ is the weight vector (or priority vector), then the elements of the matrix A = $[a_{ij}]$ can be approximated as $a_{ij} \approx \frac{w_i}{w_j}$. This means that each element of A = $[a_{ij}]$ can be expressed in terms of the ratio of weights to become A = $[\frac{w_i}{w_j}]$ or

$$A = \begin{vmatrix} 1 & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & 1 & \cdots & \frac{w_2}{w_n} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{vmatrix} \quad (2)$$

Essentially, the elements of the matrix A, as defined above, are derived from the exercise of pairwise comparisons by each decision maker. The matrix property represents this $\frac{1}{a_{ji}} = a_{ij}$. This means that if the i^th row variable of the matrix is X times more favorable to the decision maker than the corresponding j^th column variable of the matrix, then $a_{ij} = X$, while $a_{ji} = 1/X$. The total number of comparisons for each matrix of order n is n(n-1)/2.
To perform the comparisons, a lookup scale of 1 to 9 is used. Value 9 represents an extremely favored judgement, while 1 represents an extremely unfavored judgement. This range of values represents the degree of relative importance between the two variables, subjectively selected during each comparison. Saaty named this scale the 9-Point Intensity of Relative Importance Scale Table 1.

Table 1: Saaty's 9-Point Intensity of Relative Importance Scale

Intensity of Relative Importance	Definition	Explanation
1	Equal Importance	Two activities contribute equally to objective 1
3	Moderate Importance	Experience and judgment slightly favor one activity over another
5	Strong Importance	Experience and judgement slightly favor one activity over another
7	Demonstrated Importance	An activity is strongly favoured, and its dominance is demonstrated in practice
9	Extreme Importance	The evidence favouring one activity over another is of the highest possible order of affirmation
2,4,6,8	Intermediate Values Between Two Adjacent Judgements	When a compromise is needed
Reciprocals of the above nonzero numbers	Reciprocal For Inverse Comparison	

Determining the Principal Eigenvector
In the Analytic Hierarchy Process, the principal eigenvector obtained from each pairwise comparison matrix provides the priority weights of the decision criteria. These weights represent the relative importance of the criteria and form the basis for subsequent ranking and aggregation within the hierarchy. The same procedure is applied to derive the local weights of sub-criteria and, where applicable, the alternatives. To compute these weights, each pairwise comparison matrix is first normalized. This is achieved by dividing each element in a given column by the sum of all elements in that column, thereby transforming the matrix into a comparable scale. Normalization ensures that the relative contributions of individual elements are assessed consistently across the matrix. Following normalization, the priority weight for each criterion is obtained by calculating the average of the elements in the corresponding row of the normalized matrix. The resulting vector of row averages represents an approximation of the principal eigenvector associated with the maximum eigenvalue of the original comparison matrix. This eigenvector captures the relative priorities of the criteria in a form suitable for further synthesis within the AHP framework.

The mathematical formulation of these steps is presented in Equations (3) to (6).

$$A\,\omega = \begin{vmatrix} \frac{a_{11}}{\Sigma a_{i1}} & \frac{a_{12}}{\Sigma a_{i2}} & \cdots & \frac{a_{1n}}{\Sigma a_{in}} \\ \cdots & \cdots & \cdots & \cdot \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{a_{n1}}{\Sigma a_{i1}} & \frac{a_{n2}}{\Sigma a_{i2}} & \cdots & \frac{a_{nn}}{\Sigma a_{in}} \end{vmatrix} \quad (3)$$

Here is the Normalization Equation:

$$a_{ij}^{*} = \frac{a_{ij}}{\sum_{i=1}^{a} a_{ij}} \quad (4)$$

For all j = 1, 2, …, n.
Weight calculation is derived as follows:

$$\omega_{ij} = \frac{\sum_{j=1}^{n} a_{ij}^{*}}{n} \quad (5)$$

For all i = 1. 2, …, n

Performing Consistency Checks
To ensure consistency, AHP uses a consistency ratio introduced by Saaty. This ratio helps verify the reliability of the comparisons performed by each decision maker. Calculating the consistency ratio (CR) involves the following steps:

i) Calculating the weighted sum vector by multiplying the comparison matrix by the principal eigenvector.

ii) Dividing the elements of the weighted sum vector by the corresponding elements of the principal eigenvector.

These are represented mathematically in (6) and (7):

$$C = \begin{bmatrix} c_i \\ c_2 \\ c_3 \\ \cdots \\ \cdots. \\ c_n \end{bmatrix} = \begin{vmatrix} \frac{a_{11}\cdot}{\Sigma\frac{a_{11}}{n}+} & \frac{a_{12}\cdot}{\Sigma\frac{a_{12}}{n}+} & \cdots & \frac{a_{1n}\cdot}{\Sigma\frac{a_{1n}}{n}} \\ \cdots & \cdots & & \cdots. \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{a_{n1}\cdot}{\Sigma\frac{a_{n1}}{n}}+ & \frac{a_{11}\cdot}{\Sigma\frac{a_{n2+}}{n}} & \cdots & \frac{a_{11}\cdot}{\Sigma\frac{a_{nn}}{n}} \end{vmatrix} \quad (6)$$

$$A\text{x}\,C = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} \text{X} \begin{bmatrix} c_i \\ c_2 \\ c_3 \\ \cdots \\ \cdots. \\ \cdots. \\ c_n \end{bmatrix} = \begin{bmatrix} x_i \\ x_2 \\ x_3 \\ \cdots \\ \cdots. \\ \cdots. \\ x_n \end{bmatrix} \quad (7)$$

Finding the average of these values will give the maximum eigenvalue (λ_{max}) of the pairwise comparative matrix.

iii) Determining and calculating the consistency index (CI) using the formula:

$$\text{CI} = \frac{(\lambda_{max} - n)}{(n-1)} \quad (8)$$

where n is the order of the matrix.

iv) Calculating the consistency ratio (CR) by dividing the CI by the random index (RI):

$$CR = \frac{CI}{RI} \quad (9)$$

where RI is obtained from another lookup table, introduced by Saaty, called the *Random Inconsistency Index* (Table 2). Table 2 presents the RI values for matrices of orders 1 to 10.

Table 2: Saaty's Random Inconsistency Indices for n = 1, 2, …, 10

N	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
RI	*0*	*0*	*0.58*	*0.9*	*1.12*	*1.24*	*1.32*	*1.41*	*1.46*	*1.49*

If CR <= 0.1, the comparisons are acceptable and adjudged consistent. If, however, CR C >0.1, then the values of the ratio are indicative of inconsistent judgments.

In such cases, the original values in the pairwise comparison matrix A should be reconsidered and revised (Do and Kim, 2012).

To obtain an aggregate measure of the pairwise comparisons among all individuals involved in a decision problem, the geometric mean of the individual assessments, as defined in (10), is computed. (Do, and Kim, 2012)

$$a_{ij}^{hp} = \sqrt[Q]{\prod_{q=1}^{Q} \quad a_{ij}^{q}} \quad (10)$$

Where a_{ij}^{q} is an element of Matrix A of an individual group member

q (q = 1, 2, ..., Q), and $\sqrt[Q]{\prod_{q=1}^{Q} \quad a_{ij}^{q}}$ is the Geometric Mean (GM) of all individuals a_{ij}^{hp}

The geometric mean is used to aggregate the pairwise comparisons for all decision makers.

The local weights of the factors and attributes, and the consistency ratio for each matrix, are analysed using the aforementioned procedure. Global weights are synthesized from the level above by multiplying the local weights by the corresponding criterion at that level and summing the results for each element in the current level, according to the criteria it affects. (Do and Kim, 2012).

Algorithm Design – Pseudocodes and Flowcharts

A simple algorithm for adjusting an inconsistent matrix is presented as Algorithms 1 and 2, both in the form of pseudocode representing two pseudo-functions. The function *AdjustedMatrix* accepts two input parameters, namely, *criteriaCount* and *CriteriaMatrix*. The *CriteriaMatrix* is a square matrix that satisfies the conditions for the positive reciprocal matrix as defined in 2.2.1 while the *criteriaCount* is the order or size of the matrix. The *AdjustedMatrix* function returns an adjusted consistent matrix if the input *CriteriaMatrix* is inconsistent; otherwise, it returns the same matrix. Essentially, this algorithm is basically a direct repair function that takes the original matrix, aligns it step by step with the eigenvector ratios until consistency is achieved.

The second function is called *ComputeCR* (Figure 2). This function computes the consistency ratio (CR). It also accepts as input *CriteriaMatrix* and *criteriaCount.* This function is called from within the *AdjustedMatrix* function at every iteration to check if the inconsistent matrix parameter input has been adjusted to consistency. It should be noted that the core function that performs the matrix adjustment task is the *AdjustedMatrix* function. Details of other functions, and sub-functions called from this function within the *ComputerCR* function are implemented and incorporated into the workings of the C# based application designed and developed for this study.

```
FUNCTION AdjustedMatrix(criteriaCount, CriteriaMatrix)
    maxIterations = 500
    iterations = 0
    WHILE TRUE
        (priorityVector, CR) = ComputeCR(criteriaCount, CriteriaMatrix)
        IF CR <= 0.10 THEN BREAK

        FOR row = 0 TO criteriaCount-1
            FOR column = row+1 TO criteriaCount-1
                IF |CriteriaMatrix[row, column] - (priorityVector[row] / priorityVector[column])| > 0.1 THEN
                    expectedValue = priorityVector[row] / priorityVector[column]
                    CriteriaMatrix[row, column] = expectedValue
                    CriteriaMatrix[column, row] = 1 / expectedValue
                ENDIF
            NEXT column
        NEXT row

        iterations++
        IF iterations > maxIterations THEN
            PRINT "Information: CR unreachable ≤ 0.10 within max iterations."
            BREAK
        ENDIF
    ENDWHILE

    RETURN CriteriaMatrix
END FUNCTION
```

Algorithm 1: Pseudocode that represents the function *AdjustedMatrix*, whose purpose is to adjust an inconsistent AHP matrix to consistency. The function receives two parameters, namely, criteriaCount and CriteriaMatrix. criteriaCount is the matrix size, while the CriteriaMatrix is the inconsistent matrix to be adjusted by the function. Logically, within the body of the function, periodic calls is made to ComputeCR function to determine at what point the computed Consistency Ratio is not greater than 0.1. At this point, the function returns an adjusted consistent Matrix.

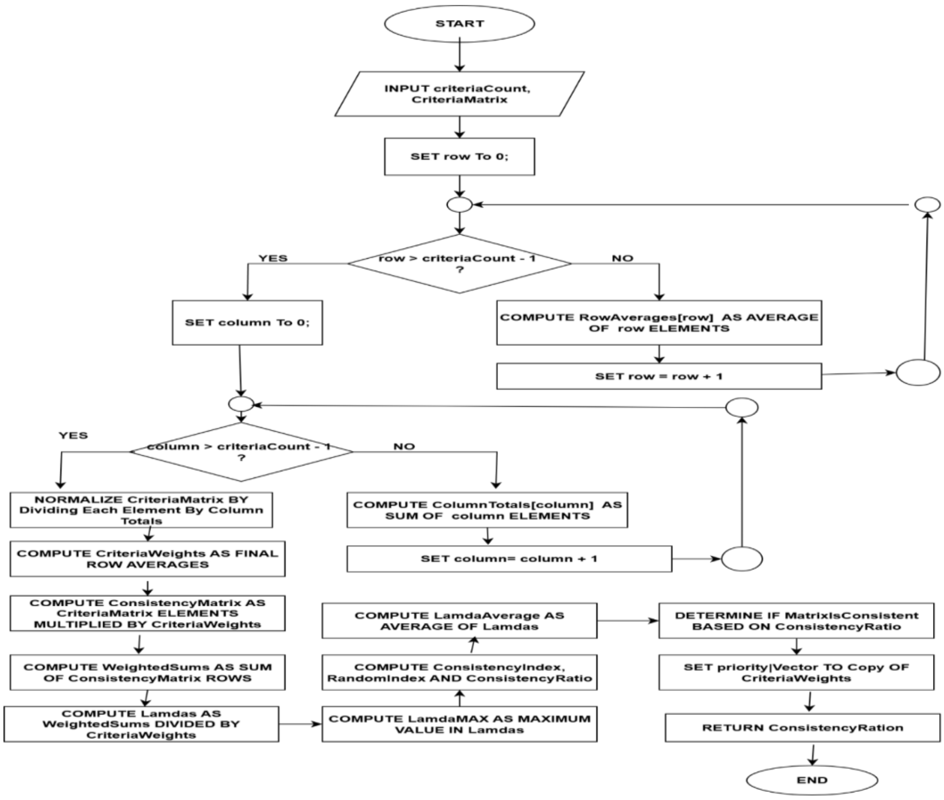

Figure 2: Flowchart Representation of the Function *ComputeCR*, whose Purpose is to Calculate and Return the Consistency Ratio (CR) of an AHP Matrix. The Function takes two Parameters: criteria Count (the Matrix Size) and Criteria Matrix (the Square Matrix), and Computes the Consistency Ratio of the Matrix

Experiments and Validation

To evaluate the behavior of the proposed automated consistency adjustment algorithm, controlled experiments were conducted using both synthetic and representative AHP judgment matrices. The evaluation focuses on the algorithm's ability to reduce inconsistency efficiently while preserving the structural characteristics of expert judgments.

Synthetic Data Generation

Synthetic AHP matrices were generated to systematically control inconsistency levels. For each experiment:

i. A ground-truth priority vector w^* with positive entries summing to one, a random number was generated.
ii. A perfectly consistent pairwise comparison matrix A^* was constructed such that
$$a_{ij}^* = \frac{w_i^*}{w_j^*}.$$
iii. Human judgment noise was simulated by perturbing off-diagonal entries with multiplicative random factors while preserving reciprocity and the Saaty scale bounds.
iv. The resulting matrices were categorized into low, medium, and high inconsistency regimes based on their initial Consistency Ratio (CR).

Each configuration was repeated over multiple random trials to ensure robustness.

Baseline Definition

The proposed algorithm was evaluated against the following baselines:

i. Unadjusted AHP baseline: original judgment matrices evaluated without modification, serving as the canonical decision-science reference.
ii. Threshold-based acceptance baseline: matrices assessed solely using the CR $\leq$ 0.10 rule, without corrective adjustment.

These baselines reflect standard practice in AHP applications and provide a meaningful point of comparison for automated consistency repair.

Evaluation Metrics

The algorithm performance was assessed using the following metrics: initial and final consistency ratio (CR), CR reduction ratio, number of iterations to convergence, computational cost, and deviation of resulting priority vectors from the underlying ground-truth priorities. The resulting evaluation results and analysis are presented in Table 3, where the consistency reduction achieved by the proposed algorithm across different noise levels is shown.

Table 3: Summary of the Achieved Consistency Reduction

Noise level	Mean CR (before)	Mean CR (after)	Max CR (after)	Mean iterations
Low	0.06	0.04	0.1	0.15
Medium	0.26	0.01	0.1	0.93
High	0.46	0.006	0.095	0.98

Across all regimes, the proposed algorithm reliably reduced CR below the commonly accepted threshold of 0.10, frequently achieving near-perfect consistency within a single iteration. This represents a substantial improvement over threshold-based acceptance without correction.

Convergence Behavior and Computational Complexity

Empirically, the algorithm converged in one iteration for most inconsistent matrices. This behavior aligns with the algorithm's fixed-point structure: once the judgment matrix becomes consistent with its derived priority vector, no further updates are required.

Each iteration involves eigenvector computation with complexity $O(n^3)$, followed by $O(n^2)$ pairwise updates. Given that convergence typically occurs within one iteration for practical AHP sizes ($n \leq 10$) The algorithm is computationally efficient for real-world decision-support systems.

Optimization-Inspired Reference Comparison

To contextualize the proposed method within the broader algorithmic literature, its behavior was contrasted with a canonical optimization-inspired consistency repair approach based on least-squares adjustment of pairwise ratios. Such methods aim to minimize deviation from original judgments subject to consistency constraints.

While least-squares formulations preserve fidelity more conservatively, they typically require iterative numerical optimization and exhibit slower convergence in highly inconsistent settings. In contrast, the proposed algorithm directly projects judgments onto a consistent structure implied by the current priority vector, achieving rapid CR reduction with minimal computational overhead. This comparison highlights complementary strengths: optimization-based approaches emphasize fidelity preservation, whereas the proposed algorithm prioritizes fast and reliable consistency enforcement.

RESULTS AND DISCUSSION

Implementation Results

The approach to implementing the proposed algorithm was based on Design Science Research (DSR). DSR is suitable because it generates domain-relevant knowledge and supports the creation of novel solutions (Wulf, 2020). The process follows six stages: problem identification, objective definition, design, demonstration, evaluation, and communication (Blanka et al., 2022). While earlier sections addressed the first two stages and algorithm design this section focuses on implementation. The developed application, using C#, was tested using empirical data collected between January and March 2025 from Delphi-AHP surveys involving e-learning experts in Nigerian Higher Education Institutions (HEIs), which generated the dataset used to construct AHP models processed by the application. The results showed that across the various stages of dataset processing and analysis, the application produced both consistent and inconsistent AHP matrices, depending on the survey respondents' judgments. A key feature of the application was the detection of an inconsistent matrix. Once an inconsistent matrix is detected, the application offers researchers the option to manually revise the initial data input by either reverting to the expert to refine their judgment or producing an equivalent consistent matrix that preserves the expert's original judgment (Table 4). Table 5 shows the summary results of the AHP judgment matrices from 16 survey responders generated from the algorithm-based web application

Table 4: An Example of Matrix Adjustment Showing Reduction of CR From 0.5824 to 0.0053 After One Iteration Produced by The Web Application

Responder		Original Inconsistent Matrix				Adjusted Consistent Matrix				
Responder1		DM1	DM2	DM3	DM4		DM1	DM2	DM3	DM4
	DM1	1	0.5	5	6	DM1	1	1.4178	7.4219	1.3238
	DM2	2	1	4	0.25	DM2	0.7053	1	5.245	0.9337
	DM3	0.2	0.25	1	0.25	DM3	0.1347	0.191	1	0.25
	DM4	0.16677	4	4	1	DM4	0.7554	1.071	4	1
		CR = 0.5824					CR = 0.0053			

Table 5: Summary of Judgment Matrices

Metric	Value
Number of respondents	16
Initially consistent matrices	3
Initially inconsistent matrices	13
Mean CR before adjustment	0.430962
Mean CR after adjustment	0.000469
Maximum CR after adjustment	0.0053
Mean iterations required	1

Discussion

In this section, we discuss some of the advantages that the proposed algorithm has over the genetic algorithm, which many authors have adapted to solving the problem of adjusting inconsistent AHP matrices to consistency. Genetic Algorithms (GAs) are optimization techniques that evolve candidate solutions across generations. They rely on populations of solutions, fitness evaluation (here, CR), selection, crossover, and mutation (Thede, 2004; Carr, 2025; Tanjani et al., 2023). Each solution (chromosome) represents parameters of the problem. Through iterative evolution, better solutions emerge based on fitness scores.

Relative Advantages of the Proposed Algorithm

Advances in science and engineering rely heavily on numerical algorithms, with multiple approaches often available for a given task (Mitsos et al., 2018). Comparing the proposed method for adjusting inconsistent AHP matrices with genetic algorithms (GAs) reveals a clear methodological distinction.

Although GAs have shown strong empirical performance, their efficiency remains debated. They are randomized optimization methods whose runtime may increase by at least a factor of ln(n), partly due to intrinsic resampling effects arising from population generation, crossover, and mutation processes (Salomon, 1999).

In contrast, this study adopts a deterministic approach to improve AHP matrix consistency. GAs are typically population-based, rely on fitness optimization (balancing consistency ratio and closeness to original values), and produce stochastic outcomes across iterations (Tanjani et al., 2023). The proposed algorithm offers several advantages: 1- it is deterministic as it converges directly to a consistent matrix via the priority vector, 2 – it is simpler as it avoids population, crossover, and mutation processes, 3 – it is faster as it achieves convergence in a few iterations, 4 – it is a direct repair method as it incrementally aligns matrix entries with eigenvector ratios and it is practical in the sense that it iseasier to implement in C#, requiring only a single matrix rather than multiple candidate populations

CONCLUSION

The key contribution of this deterministic algorithm lies in its alignment with real-world AHP practice, where each expert provides a single matrix. When inconsistency arises, the matrix is either revised or automatically adjusted, resulting in one consistent matrix per expert. Unlike GA-based methods, which require multiple population candidates, the proposed approach operates efficiently on a single matrix.

The algorithm was validated through a web-based C# application using empirical Delphi-AHP data from a case study on prioritizing e-learning challenges in Nigerian Higher education institutions. The results demonstrate the effectiveness of automating consistency adjustment, particularly when experts are unable or unwilling to revise their judgments.

Given its domain-independent applicability, future work may integrate Fuzzy AHP to address uncertainty and extend the framework to Delphi Fuzzy-AHP models. Additionally, hybrid approaches combining deterministic and genetic methods could further enhance performance.

REFERENCES

Amenta, P., Lucadamo, A., & Marcarelli, G. (2021). On the choice of weights for aggregating judgments in non-negotiable AHP group decision making. *European Journal of Operational Research*, *288*(1), 294–301. doi: https://doi.org/10.1016/j.ejor.2020.05.048

Blanka, C., Krumay, B., & Rueckel, D. (2022). The interplay of digital transformation and employee competency: A design science approach. *Technological Forecasting and Social Change*, *178*, 121575. doi: https://doi.org/10.1016/j.techfore.2022.121575

Carr, J. (n.d.). *An Introduction to Genetic Algorithms*.
editor, P. R. F. B. P. R. is a copy, Economics, F.-C. with E. in, Finance, P., & policies, over twenty years of experience in the classroom L. about our editorial. (n.d.). *What Is the Delphi Method, and How Is It Useful in Forecasting?* Investopedia. Retrieved September 5, 2025, from https://www.investopedia.com/terms/d/delphi-method.asp

Farid, S., Ahmad, R., & Alam, M. (2015). A Hierarchical Model for E-learning Implementation Challenges using AHP. *Malaysian Journal of Computer Science*, *28*(3), Article 3.

Han, E. (2021). The Application of Delphi-AHP Method in the Priority of Policies for Expanding the Use of Artificial Intelligence. *Journal of Internet Computing and Services*, *22*(4), 99–110. doi: https://doi.org/10.7472/jksii.2021.22.4.99

Humphrey-Murto, S., & Wit, M. de. (2019). The Delphi method—More research please. *Journal of Clinical Epidemiology*, *106*, 136–139. doi: https://doi.org/10.1016/j.jclinepi.2018.10.011

Li, Q., Yin, X., Meng, S., Liu, Y., & Ying, Z. (2020). A security event description of intelligent applications in edge-cloud environment. *Journal of Cloud Computing*, *9*(1), Article 1. doi: https://doi.org/10.1186/s13677-020-00171-0

Liu, Z.-L., Liu, F., Zhang, J.-W., & Pedrycz, W. (2023). Optimizing consistency and consensus in group decision making based on relative projection between multiplicative reciprocal matrices. *Expert Systems with Applications*, *224*, 119948. doi: https://doi.org/10.1016/j.eswa.2023.119948

Madzík, P., & Falát, L. (2022). State-of-the-art on analytic hierarchy process in the last 40 years: Literature review based on Latent Dirichlet Allocation topic modelling. *PLOS ONE*, *17*(5), e0268777. doi: https://doi.org/10.1371/journal.pone.0268777

Mitsos, A., Najman, J., & Kevrekidis, I. G. (2018). Optimal deterministic algorithm generation. *Journal of Global Optimization*, *71*(4), 891–913. doi: https://doi.org/10.1007/s10898-018-0611-8

Muhammad, A., Shaikh, A., Naveed, Q. N., & Qureshi, M. R. N. (2020). Factors Affecting Academic Integrity in E-Learning of Saudi Arabian Universities. An Investigation Using Delphi and AHP. *IEEE Access*, *8*, 16259–16268. IEEE Access. doi: https://doi.org/10.1109/ACCESS.2020.2967499

Pant, S., Kumar, A., Ram, M., Klochkov, Y., & Sharma, H. K. (2022). Consistency Indices in Analytic Hierarchy Process: A Review. *Mathematics*, *10*(8), 1206. doi: https://doi.org/10.3390/math10081206

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (Ninth edition). McGraw-Hill Education.

Saaty, R. W. (1987). The analytic hierarchy process—What it is and how it is used. *Mathematical Modelling*, *9*(3), 161–176. doi: https://doi.org/10.1016/0270-0255(87)90473-8

Saaty, T. L. (1977). A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, *15*(3), 234–281. https://doi.org/10.1016/0022-2496(77)90033-5

Saaty, T. L., & Vargas, L. G. (2012). Why is the Principal Eigenvector Necessary? In T. L. Saaty & L. G. Vargas (Eds.), *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process* (pp. 63–70). Springer US. https://doi.org/10.1007/978-1-4614-3597-6_4

Salomon, R. (1999). The deterministic genetic algorithm: Implementation details and some results. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, *1*, 695-702 Vol. 1. doi: https://doi.org/10.1109/CEC.1999.782001

Sommerville, I. (2002). *Software engineering* (6. ed., [Nachdr.]). Addison-Wesley.

Sorooshian, S. (2015). *Analytic Hierarchy Process Decision Making Algorithm*. doi: https://doi.org/10.13140/RG.2.1.2175.6880

Tajani, Z., Faculty of Sciences, Moulay Ismail University, Tajani, C., Polydisciplinary Faculty of Larache, Abdelmalek Essaadi University, Khattabi, I., Polydisciplinary Faculty of Larache, Abdelmalek Essaadi University, Sabbane, M., & Faculty of Sciences, Moulay Ismail University. (2023). A metaheuristic approach to improve consistency of the pairwise matrix in AHP. *Mathematical Modeling and Computing*, *10*(4), 1164–1173. doi: https://doi.org/10.23939/mmc2023.04.1164

Thede, S. M. (2004). *AN INTRODUCTION TO GENETIC ALGORITHMS*.