

## A metaheuristic approach to improve consistency of the pairwise matrix in AHP

Tajani Z.<sup>1</sup>, Tajani C.<sup>2</sup>, Khattabi I.<sup>2</sup>, Sabbane M.<sup>1</sup>

<sup>1</sup>*Faculty of Sciences, Moulay Ismail University,  
Avenue Zitoune, 11201, Meknes, Morocco*

<sup>2</sup>*Polydisciplinary Faculty of Larache, Abdelmalek Essaadi University,  
745 Omar Ben Abdelaziz Avenue, 92004, Larache, Morocco*

(Received 20 August 2023; Revised 2 November 2023; Accepted 3 November 2023)

In this paper, we are interested in modifying inconsistent pairwise comparison matrix which is a critical step in the AHP methodology, where decision makers have to improve the consistency by revising the process. To this end, we propose an improved genetic algorithm (GA) to allow decision makers to find an appropriate matrix and adjust the consistency of their judgment without loss of original comparison matrix. Numerical results with different dimensions of matrices taken randomly show the effectiveness of these strategy to improve and identify the consistency of pairwise matrix which mean that GAs are a very good tool to generate the consistent pairwise comparison matrices with different number of criteria.

**Keywords:** *genetic algorithm; pairwise matrix; analytic hierarchy process; decision theory; consistency.*

**2010 MSC:** 68W20, 15Axx, 90Bxx, 90B50, 91B06     **DOI:** 10.23939/mmc2023.04.1164

### 1. Introduction

Analytic Hierarchy Process (AHP) is a multi-attribute decision-making methodology developed by Saaty [1, 2]. It is widely used in a variety of decision situations, in social, economic, military, management, and many other fields [3–5].

The main problem of the AHP method in the practical application is the judgment matrix which is constructed only by decision-makers based on their experience and knowledge. Thereby, the pairwise comparison matrix could be inconsistent due to the limitations of experiences and expertise as well as the complex nature of the decision problem [6, 7], especially when dealing with a great number of judgments. Saaty [8] proposed the consistency ratio  $CR$  to test whether the matrix is consistent. Indeed, the pairwise comparison matrix can pass the consistency test, if the consistency ratio  $CR < 0.1$ . However, in many cases, the judgment matrix cannot pass this test to be acceptable and it needs to be adjusted.

The consistency of pairwise matrix has been a subject of many studies for several decades [9–12]. However, some developed methods are complicated and difficult to use in the revising process of the inconsistent comparison matrix while some are difficult to preserve most of the original comparison information since a new matrix has to be constructed to replace the original comparison matrix.

Genetic algorithms (GAs) are a class of metaheuristic approaches, developed by Jean Holland [13]. They are inspired by the natural biological mechanisms of the theory of evolution, proposed by Charles Darwin [14]. It has been extensively employed to solve several problems [15–18]. It consists of multiple iterative processes which makes it possible to get closer and closer to the optimal solution. To benefit from the advantages of GAs as fundamental approaches and to improve their performance, it is necessary to adapt and adopt suitable genetic operators for the resolution of the considered problem.

The aim of this paper is to propose an alternative approach to create consistent pairwise matrices. This approach is based on GA in defining a set of consistent pairwise matrices for different dimensions

to aid experts to define the appropriate judgment decision. Thus, instead of determining a pairwise matrix that can be inconsistent, especially in the case of high order, and trying to modify it, the expert will have pairwise comparison matrices during the algorithm process allowing the identification of the appropriate one.

The remaining parts of this paper are organized as follows. In the second section, we briefly describe the different steps of the Analytic Hierarchy Process and the problem of inconsistency in the pairwise comparison matrix. In Section 3, we present the genetic approach to identify consistent matrices and modify an inconsistent judgment matrix. In Section 4, numerical results are performed with different sizes of matrices showing the effectiveness and accuracy of the proposed method to overcome the problem of inconsistency in pairwise comparison matrix in the AHP method.

## 2. Problem of pairwise comparison matrix

In Analytic Hierarchy Process (AHP), the process of decision-making starts with breaking down the multi-criteria decision-making problem into a hierarchy model, then we can find the weights by using mathematical calculation based on linear algebra. These weights can be generated with the help of the pairwise comparisons of two alternatives under the given criterion.

Pairwise comparison is a fundamental concept in the Analytic Hierarchy Process (AHP). It involves systematically comparing elements in a given level with respect to a shared criterion or property in the level above. Decision matrices resulting from an application of AHP can be considered an effective method to structure and represent relevant information of a strategic problem.

The pairwise comparison matrix has the form defined below:

$$A = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1n} \\ a_{21} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & 1 \end{pmatrix}.$$

The matrix  $A = (a_{ij})$ ,  $i, j = 1, \dots, n$  has a positive entries everywhere and satisfies the reciprocal property  $a_{ji} = 1/a_{ij}$ . The comparisons are made using a scale that indicates the importance of one element over another element with respect to a given attribute.

Table 1 shows a scale ranging from 1 for ‘least valued than’ to 9 for ‘definitely most important than’.

**Table 1.** 1–9 Scale for the pairwise comparison (Saaty 2001).

| Linguistic term           | Preference number |
|---------------------------|-------------------|
| Equally important         | 1                 |
| Weakly more important     | 3                 |
| Strongly more important   | 5                 |
| Very strong important     | 7                 |
| Absolutely more important | 9                 |
| Intermediate values       | 2, 4, 6, 8        |

Inconsistencies can arise, especially when dealing with a large number of judgments, since pairwise judgments are based primarily on the experience and knowledge of experts. Hence the need to use a method capable of evaluating this consistency in a specific problem.

The pairwise comparison matrix needs to pass a consistency test to be used. Saaty developed a ‘consistency ratio’ as a consistency test that allows a certain level of acceptable deviations. It is defined as follows:

$$CR = \frac{CI}{RI},$$

where

$$CI = \frac{\lambda_{\max} - n}{n - 1},$$

$\lambda_{\max}$  is the maximum eigenvalue of the pairwise comparison matrix, and  $RI$  is a random index, whose value depends on  $n$ , given in Table 2.

**Table 2.** Random Consistency Index.

|      |   |   |      |     |      |      |      |      |      |      |
|------|---|---|------|-----|------|------|------|------|------|------|
| $n$  | 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 | 1.45 | 1.49 |

If  $CR > 0.1$ , the decision-maker is asked to revise his judgments until an acceptable level of consistency is reached.

### 3. Approach genetic for pairwise matrix

#### 3.1. Overview of genetic algorithms

Genetic algorithms (GAs), which are primarily developed by Holland, have proven to be effective in solving a variety of optimization problems. They are based on the principles of biological evolution and operate as a searching method. A population of chromosomes is used to represent potential solutions, and genetic operators are applied to progressively improve each chromosome, which becomes the basis for the next generation. This process continues until the desired number of generations has been completed or a predefined fitness value has been reached.

GAs offer a number of advantages over other optimization approaches. First, they search from a population of solutions instead of just one. Second, they can use any fitness function, even if it is not continuous. Third, they use random operators to generate new solutions. Fourth, they do not need to know anything about the problem to find a good solution.

GAs are based on genetic operators (selection, crossover, mutation, ...) which must be adapted following the nature of the problem. In this paper, we describe the steps of an improved genetic algorithm for the considered problem with adapted operators.

#### 3.2. Genetic operators

The implementation of genetic algorithm to solve a given problem, goes through an important step consisting in representing the form of possible solutions of the problem treated. According to the representation or the type of coding, which can be binary, real, matrix, prefer number, ... we must adapt and apply a suitable operators which will allow to approach the sought solution and at the same time perform the quality of the proposed algorithm. Concerning our problem we are opted for the genetic operators defined bellows knowing that we have performed different choice of operators. Genetic algorithms typically consist of the following basic elements:

##### 3.2.1. Initialization

Genetic algorithms begins by creating a population of potential solutions to the problem being solved. This is typically done by randomly generating a set of individuals, where each individual is a potential solution represented as a set of genes or chromosomes.

In our case, the pairwise matrix, which is the objective of the problem, can be identified by  $\frac{n(n-1)}{2}$  elements from the set  $\{2, 3, \dots, 9, 1/2, 1/3, \dots, 1/9\}$ . Thus, we have to generate randomly  $m$  matrices, each of them is encoded by a vector under the form:

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

**Example:**

$$\boxed{2 \mid 1/3 \mid 5 \mid 4 \mid 1/7 \mid 2} \Rightarrow \begin{pmatrix} 1 & 2 & 1/3 & 5 \\ 1/2 & 1 & 4 & 1/7 \\ 3 & 1/4 & 1 & 2 \\ 1/5 & 7 & 1/2 & 1 \end{pmatrix}$$

### 3.2.2. Fitness function

The fitness function is used to evaluate each individual in the population and assign a fitness score based on how well it solves the problem being considered. The fitness score is used to select individuals for reproduction in the next generation. The fitness function is defined as  $CR$ , which is the consistency index function.

### 3.2.3. Selection

The selection process involves choosing the fittest individuals from the current generation to be parents for the next generation. The individuals are selected using various techniques such as roulette wheel selection or tournament selection.

### 3.2.4. Crossover

It is the process of combining genetic material from two parents to create a new individual in the next generation. This operator must be adapted following the considered encoding. In the pairwise comparison matrix problem, we consider three crossover operators:

- One point crossover: In this approach, a random position or index in the parent chromosomes is selected, and the genetic material beyond that point is exchanged between the parents, producing two offspring.

**Example:**

|         |     |     |   |   |     |     |        |     |     |   |   |     |     |
|---------|-----|-----|---|---|-----|-----|--------|-----|-----|---|---|-----|-----|
| Parent1 | 2   | 1/5 | 7 | 3 | 1/9 | 1/2 | Child1 | 2   | 1/5 | 7 | 2 | 4   | 1/4 |
| Parent2 | 1/2 | 7   | 8 | 2 | 4   | 1/4 | Child2 | 1/2 | 7   | 8 | 3 | 1/9 | 1/2 |

- Two point crossover: It is similar to single point crossover, but instead of a single point, two points are selected. The genetic material between the two points is exchanged between the parents.

**Example:**

|         |     |     |   |   |     |     |        |     |     |   |   |     |     |
|---------|-----|-----|---|---|-----|-----|--------|-----|-----|---|---|-----|-----|
| Parent1 | 2   | 1/5 | 7 | 3 | 1/9 | 1/2 | Child1 | 2   | 1/5 | 8 | 2 | 1/9 | 1/2 |
| Parent2 | 1/2 | 7   | 8 | 2 | 4   | 1/4 | Child2 | 1/2 | 7   | 7 | 3 | 4   | 1/4 |

- Uniform crossover: We do not divide the chromosome into segments, rather we treat each gene separately. In this, we basically flip a coin for each chromosome to decide whether or not it will be included in the offspring. In other word, we consider a binary mask where the 1-bits are uniformly distributed in a binary chromosome. Thus, the cross between parent 1 and parent 2 gives rise to two children where children 1 contains the same elements of parent 1 by taking the elements of parent 2 located at the same location of the considered binary chromosome. The children 2 is created by the same procedure. An example of the uniform crossover is presented bellow:

**Example:**

|         |     |     |   |   |     |     |
|---------|-----|-----|---|---|-----|-----|
| Parent1 | 2   | 1/5 | 7 | 3 | 1/9 | 1/2 |
| Parent2 | 1/2 | 7   | 8 | 2 | 4   | 1/4 |
| mask    | 0   | 0   | 1 | 0 | 1   | 1   |
| Child1  | 2   | 1/5 | 8 | 3 | 4   | 1/4 |
| Child2  | 1/2 | 7   | 7 | 2 | 1/9 | 1/2 |

### 3.2.5. Mutation

The mutation operator involves introducing random changes or mutations in individuals to maintain diversity and explore new regions of the solution space. And that is why we used Mutation by insertion, which consists in selecting one or more random genes which will be replaced by new elements. An example of insertion mutation is presented bellow:

**Example:**

$$\boxed{2 \quad 1 \quad 1/2 \quad 7 \quad 1/3} \Rightarrow \boxed{2 \quad 1 \quad 1/5 \quad 7 \quad 1/3}$$

### 3.2.6. Elitism

In the introduction step, we keep 10% of the best chromosomes from the previous population in the new population to make sure they do not move out of solution, and we replace the remaining 90% with the new generation.

### 3.2.7. Termination

The algorithm terminates when a stopping criteria is met, such as reaching desired fitness score or running for a certain number of generations. These elements work together to produce a population of increasingly fit individuals that can be used to find optimal solutions to a wide range of problems.

## 3.3. The proposed approach

The proposed algorithm, as illustrated in Figure 1, is based on an improved genetic algorithm to find an acceptable pairwise matrix with different sizes, depending on the studied application, to help experts to have a panel of choices of acceptable matrices to define by their expertise the desired matrix or to complete an incomplete defined pairwise matrix.

The proposed approach is a combination of an adapted GA and the AHP method. GA method aim to exploit the space of possible solutions through different genetic operators; however, the AHP method will make it possible to evaluate the acceptability of the matrices by calculating their consistency ratio  $CR$  until a stopping test is satisfied. It should be noted that following the desired results, two stopping criteria are considered: The first one consists to get  $CR < 0.1$  and the second one is the maximum number of iterations.

The different steps of the proposed approach is described bellow.

---

**Algorithm 1** GA-based approach to identify consistent pairwise comparison matrix.

---

- 1: **Step 1** Define the number of criteria  $n$ ;
  - 2: **Step 2** Initialization: random generation of initial population  $V^{(0)}$  of  $k$  vectors  $(V_i^{(0)})$ ,  $i = 1, \dots, k$ ; of length  $n(n-1)/2$  from the set  $\{2, 3, \dots, 9, 1/2, 1/3, \dots, 1/9\}$ ;
  - 3: **Step 3** Encoding: transform each vector of  $V^{(0)} = (V_i^{(0)})$  to a population of pairwise matrices  $M^{(0)} = (M_i^{(0)})$  with  $i = 1, \dots, k$ ;
  - 4: **Step 4** Evaluation: calculate  $CR(i)$  for each matrix  $M_i^{(0)}$  for  $i = 1, \dots, k$ ;
  - 5: **Step 5** Genetic process: generate  $V^{(1)} = M_u \cdot C_r \cdot S_e(V^{(0)})$   
with  $S_e$  roulette selection;  
 $C_r$  crossover operator with probability  $p_c$ ;  
 $M_u$  mutation operator with probability  $p_m$ .
  - 6: **Step 6** Repeat the step 4 with  $M^{(1)}$  replace  $M^{(0)}$ ;
  - 7: **Step 7** The process continue until a stopping test is satisfied.
-

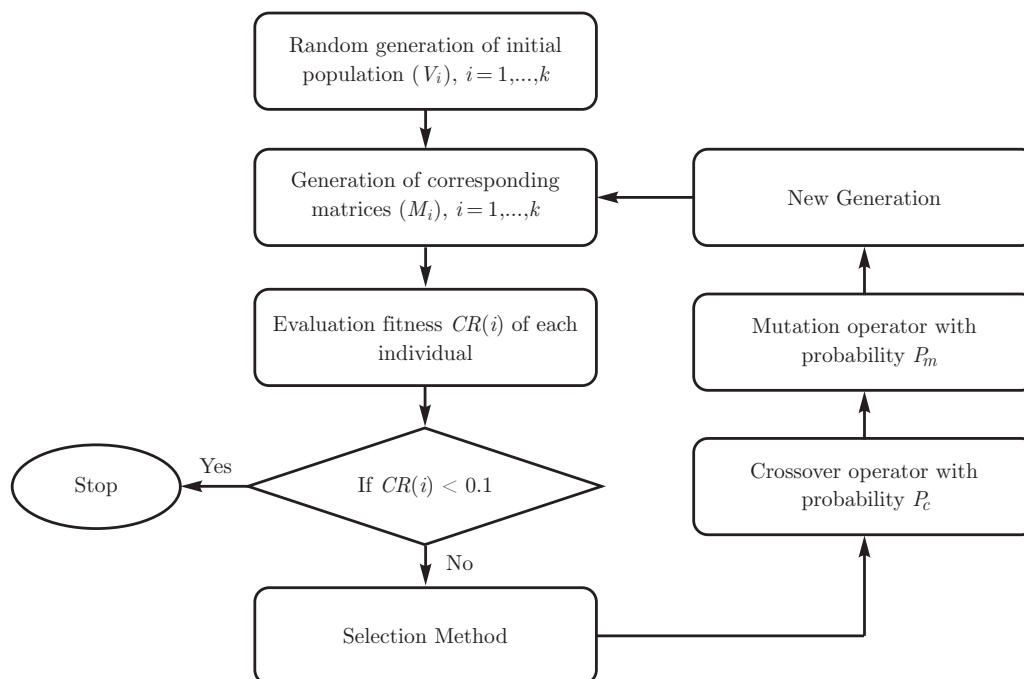


Fig. 1. Schema of the proposed approach.

#### 4. Numerical results and discussion

In order to show the efficiency of genetic algorithms and their ability to provide experts with a number of acceptable matrices ( $CR < 0.1$ ) in a reasonable time, numerical experiments are developed using an Intel(R) Core(TM) i3-6006U CPU @ 2.00 GHz RAM 4.00 GB.

Several simulations were performed to highlight the ability and efficiency of GAs to identify consistent matrices with different numbers of criteria, which gives rise to matrices with different sizes. Thus, the size of the population and the maximum number of iterations have been adapted according to the size of the matrix to be identified.

The considered genetic operators and corresponding parameters are described bellow:

Selection: Roulette selection

Crossover: One point crossover, two point crossover, uniform crossover, with the probability  $p_c = 0.9$

Mutation: Insertion mutation, with the probability  $p_m = 0.05$

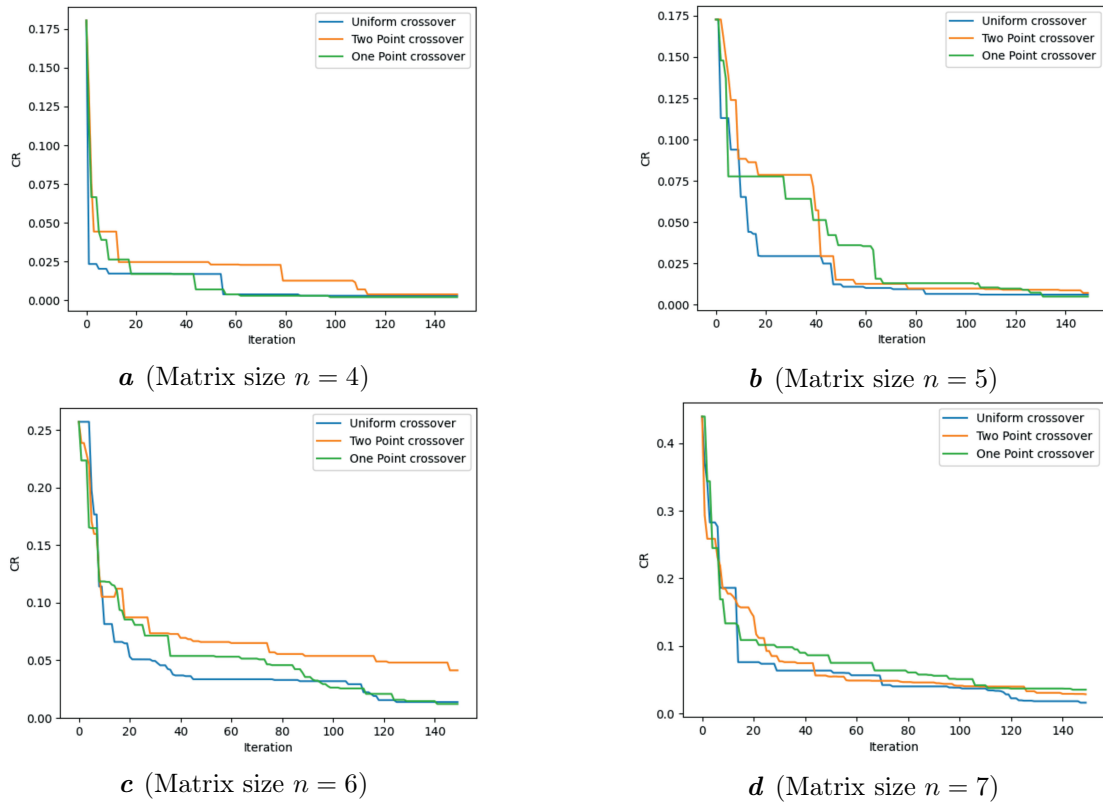
Insertion: 10% are conserved for the next generation.

It is well known that genetic operators influence the performance of GAs, in particular, the crossover operator. Thus, we have exploited three crossover operators; namely, one-point crossover, two-point crossover and uniform crossover.

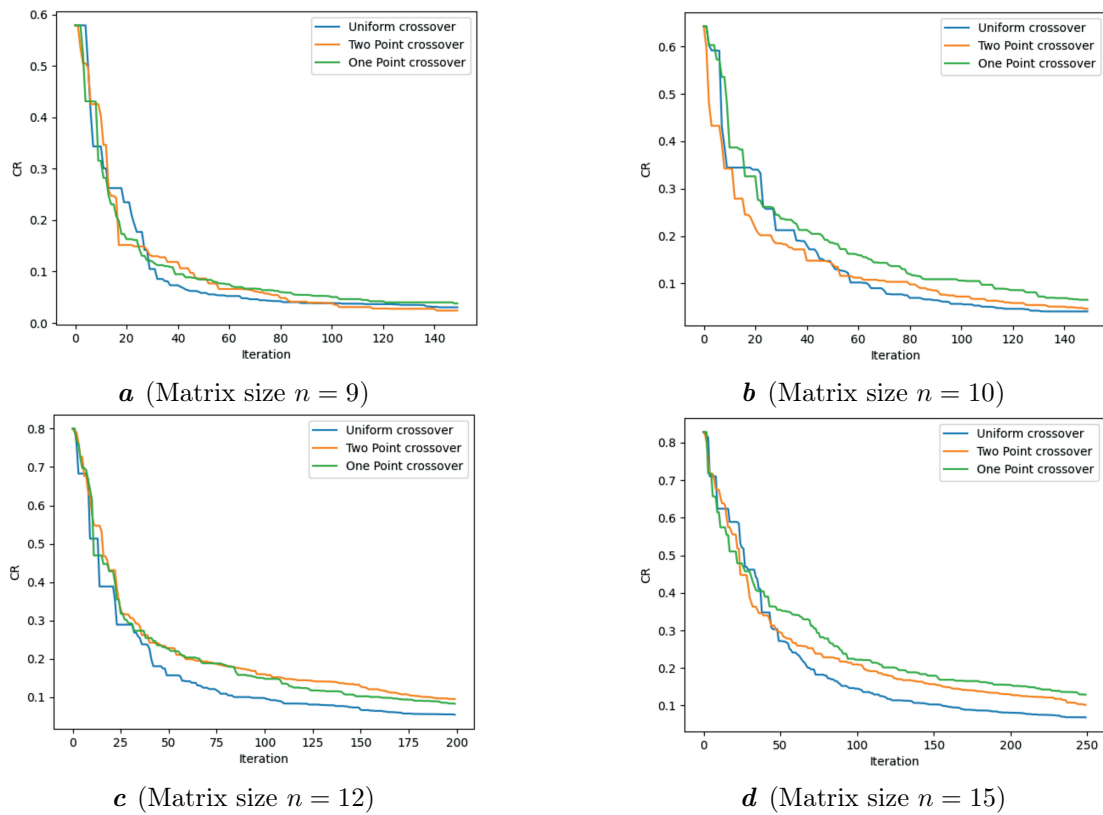
Figures 2 and 3 present the evolution of  $CR$  during the iterative process for different number of criteria 4, 5, 6, 7, 9, 10, 12 and 15 with different choice of crossover operators. It shows that the proposed algorithm with a uniform crossover produces a better evolution of  $CR$  in less iterations.

Tables 3 and 4 present a comparison of the performance of different crossover operators in GA to produce acceptable pairwise matrix. Particular, we present the required number of iteration to obtain a consistent pairwise matrix with  $CR < 0.1$ .

The results obtained show that the algorithm proposed with the three types of crossover makes it possible to have a suitable pairwise matrix after a very small number of iterations, which can increase according to the order of the matrix. In particular, the uniform crossover operator requires fewer iterations to arrive at a first matrix with the requested criterion. In addition, during the iterative process, the  $CR$  continues to decrease, which allows experts to make their choices from one iteration to another according to their data.



**Fig. 2.** Comparison of the performance of different crossover operators in GA to produce consistent pairwise Matrix.



**Fig. 3.** Comparison of the performance of different crossover operators in GA to produce consistent pairwise Matrix.

**Table 3.** Comparison of the performance of different crossover operators in GA with 4, 5, 6 and 7 criteria.

| Criteria            | 4         |            | 5         |            | 6         |            | 7         |            |
|---------------------|-----------|------------|-----------|------------|-----------|------------|-----------|------------|
| Crossover           | <i>CR</i> | Iterations | <i>CR</i> | Iterations | <i>CR</i> | Iterations | <i>CR</i> | Iterations |
| Uniform crossover   | 0.023     | 1          | 0.094     | 6          | 0.082     | 10         | 0.076     | 14         |
| One point crossover | 0.076     | 2          | 0.088     | 9          | 0.087     | 18         | 0.093     | 15         |
| Two point crossover | 0.067     | 2          | 0.078     | 5          | 0.094     | 16         | 0.098     | 29         |

**Table 4.** Comparison of the performance of different crossover operators in GA with 9, 10, 12 and 15 criteria.

| Criteria            | 9         |            | 10        |            | 12        |            | 15        |            |
|---------------------|-----------|------------|-----------|------------|-----------|------------|-----------|------------|
| Crossover           | <i>CR</i> | Iterations | <i>CR</i> | Iterations | <i>CR</i> | Iterations | <i>CR</i> | Iterations |
| Uniform crossover   | 0.086     | 32         | 0.098     | 64         | 0.099     | 93         | 0.098     | 101        |
| One point crossover | 0.097     | 45         | 0.098     | 80         | 0.099     | 186        | 0.099     | 240        |
| Two point crossover | 0.095     | 39         | 0.097     | 108        | 0.099     | 160        | 0.098     | 258        |

**Table 5.** Numerical experiment with different matrices.

| Matrix size | Population | Initial <i>Min_CR</i> | Initial <i>Max_CR</i> | <i>CR</i> < 0.1 | Iterations |
|-------------|------------|-----------------------|-----------------------|-----------------|------------|
| 4           | 20         | 0.023                 | 2.637                 | 0.023           | 1          |
| 5           | 40         | 0.173                 | 1.819                 | 0.094           | 6          |
| 6           | 60         | 0.257                 | 1.541                 | 0.082           | 10         |
| 7           | 80         | 0.371                 | 1.460                 | 0.076           | 14         |
| 8           | 100        | 0.252                 | 1.488                 | 0.095           | 28         |
| 9           | 120        | 0.579                 | 1.515                 | 0.086           | 32         |
| 10          | 140        | 0.643                 | 1.407                 | 0.098           | 64         |
| 12          | 160        | 0.800                 | 1.251                 | 0.099           | 93         |
| 15          | 200        | 0.829                 | 1.282                 | 0.098           | 101        |

**Table 6.** Evolution of percentage of consistent matrices with different criteria.

| Criteria | <i>Min_CR</i> | <i>Max_CR</i> | % of matrices with <i>CR</i> < 0.1 | Iterations |
|----------|---------------|---------------|------------------------------------|------------|
| 5        | 0.173         | 1.819         | 0%                                 | 1          |
|          | 0.065         | 0.942         | 42.5%                              | 10         |
|          | 0.030         | 0.756         | 87.5%                              | 20         |
| 7        | 0.371         | 1.460         | 0%                                 | 1          |
|          | 0.076         | 1.091         | 3.75%                              | 20         |
|          | 0.064         | 0.390         | 57.5%                              | 40         |
| 9        | 0.057         | 0.673         | 81.25%                             | 60         |
|          | 0.579         | 1.515         | 0%                                 | 1          |
|          | 0.235         | 0.953         | 0%                                 | 20         |
| 12       | 0.073         | 0.457         | 25%                                | 40         |
|          | 0.052         | 0.299         | 86.66%                             | 60         |
|          | 0.800         | 1.251         | 0%                                 | 1          |
| 12       | 0.109         | 0.380         | 0%                                 | 80         |
|          | 0.097         | 0.306         | 20%                                | 100        |
|          | 0.079         | 0.210         | 78%                                | 130        |

To better perform the GA, we have to adapt the considered number of population to get consistent pairwise matrix in less iterations. Table 5 presents the number of necessary iterations to obtain a first pairwise matrix with *CR* < 0.1 corresponding to a random generated matrices having a better individual in terms of consistency with a *CR* greater than 0.1.

It should be noted that the numerical simulations were carried out with random examples where all the matrices of the initial population are inconsistent (with *CR* > 0.1). In this regard, the rate



of consistent matrices goes from 0% for all the examples considered and gradually increases after a reduced number of iterations to reach a very high rate of consistent matrices. In other words, the experts will have a set containing an increasing number of consistent matrices as the genetic process progresses to find a pairwise consistent matrix and thus avoid inconsistent matrices (see Table 6).

The study carried out aims to show the importance and efficiency of GAs in producing consistent pairwise matrices according to the criteria given by Saaty. So, GAs present an interesting method to help experts to define the desired and appropriate matrix according to the application and the objective of their study.

## 5. Conclusion

In this work, we have exploited genetic algorithms (GAs) as a metaheuristic method for the determination of consistent pairwise matrices, when we use the AHP method. The different results obtained for matrices with different sizes have shown that GAs can be considered as a good alternative to help experts to have a set of choices of consistent matrices to consider or to define incomplete pairwise matrices. In addition, the performance of GAs can be improved, when adapting different genetic operators, to generate pairwise consistent matrices, especially for large size matrices in an acceptable time.

- 
- [1] Saaty T. L. *The Analytic Hierarchy Process*. McGraw-Hill, New York (1980).
  - [2] Saaty T. L. Homogeneity and clustering in AHP ensures the validity of the scale. *European Journal of Operational Research*. **72** (3), 598–601 (1994).
  - [3] Breaz R. E., Bologa O., Racz S. G. Selecting industrial robots for milling applications using AHP. *Procedia Computer Science*. **122**, 346–353 (2017).
  - [4] Greiner M. A., Fowler J. O., Shunk D. L., Carlyle W. M., McNutt R. T. A hybrid approach using the analytic hierarchy process and integer programming to screen weapon systems projects. *IEEE Transactions on Engineering Management*. **50** (2), 192–203 (2003).
  - [5] Vaidya O. S., Kumar S. Analytic hierarchy process: An overview of applications. *European Journal of Operational Research*. **169** (1), 1–29 (2006).
  - [6] Lance E. F., Verdini W. A. A consistency test for AHP decision makers. *Decision Sciences*. **20** (3), 575–590 (1989).
  - [7] Murphy C. K. Limits on the Analytic Hierarchy Process from its consistency index. *European Journal of Operational Research*. **65** (1), 138–139 (1993).
  - [8] Saaty T. L. Some mathematical concepts of the analytic hierarchy process. *Behaviormetrika*. **18** (29), 1–9 (1991).
  - [9] Ergu D., Kou G., Peng Y., Shi Y. A simple method to improve the consistency ratio of the pair-wise comparison matrix in ANP. *European Journal of Operational Research*. **213** (1), 246–259 (2011).
  - [10] Cao D., Leung L. C., Law J. S. Modifying inconsistent comparison matrix in analytic hierarchy process: A heuristic approach. *Decision Support Systems*. **44** (4), 944–953 (2008).
  - [11] Linares P. Are inconsistent decisions better? An experiment with pairwise comparisons. *European Journal of Operational Research*. **193** (2), 492–498 (2009).
  - [12] Saaty T. L. Decision-making with the AHP: Why is the principal eigenvector necessary. *European Journal of Operational Research*. **145** (1), 85–91 (2003).
  - [13] Goldberg D. E. *Genetic Algorithm in Search, Optimisation and Machine Learning*. Addison-Wesley, Reading (1989).
  - [14] Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin (1996).
  - [15] Mera N. S., Elliott L., Ingham D. B. A Real Coded Genetic Algorithm Approach for Detection of Subsurface Isotropic and Anisotropic Inclusions. *Inverse Problems in Engineering*. **11** (2), 157–173 (2003).

- [16] Jouilik B., Tajani C., Daoudi J., Abouchabaka J. Numerical Optimization Algorithm Based On Genetic Algorithm For A Data Completion Problem. *TWMS Journal of Applied and Engineering Mathematics*. **13** (1), 86–97 (2023).
- [17] Costa J. F. D. S. A Genetic Algorithm to Obtain Consistency in Analytic Hierarchy Process. *Brazilian Journal of Operations & Production Management*. **8** (1), 55–64 (2011).
- [18] Fei L., Guangzhou Z. Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Systems with Applications*. **36** (3), 6995–7001 (2009).

## Метаевристичний підхід для покращення узгодженості парної матриці в АНР

Таяні З.<sup>1</sup>, Таяні К.<sup>2</sup>, Хаттабі І.<sup>2</sup>, Саббане М.<sup>1</sup>

<sup>1</sup>*Факультет наук, Університет Мулая Ісмаїла, Авеню Зітун, 11201, Мекнес, Марокко*

<sup>2</sup>*Полідисциплінарний факультет Лараши, Університет Абдельмалека Ессааді, 745 Омар Бен Абделазіз Авеню, 92004, Лараши, Марокко*

У цій статті ми зацікавлені в модифікації неузгодженої матриці парного порівняння, яка є критичним кроком у методології АНР, де особи, які приймають рішення, мають покращити узгодженість шляхом перегляду процесу. З цією метою пропонується покращений генетичний алгоритм (GA), щоб дозволити особам, які приймають рішення, знаходити відповідну матрицю та коригувати узгодженість свого судження без втрати вихідної матриці порівняння. Числові результати з матрицями різних розмірів, узятих випадково, виявляють ефективність цієї стратегії для покращення та визначення узгодженості попарної матриці, що означає, що GA є дуже хорошим інструментом для створення узгоджених матриць попарного порівняння з різною кількістю критеріїв.

**Ключові слова:** *генетичний алгоритм; попарна матриця; аналітичний процес ієрархії; теорія прийняття рішень; послідовність.*