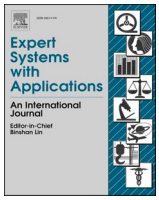
Expert Systems With Applications 245 (2024) 122973

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Neutrosophic genetic algorithm and its application in clustering analysis of rock discontinuity sets

Rui Yong, Hanzhong Wang, Jun Ye *, Shigui Du, Zhanyou Luo

Institute of Rock Mechanics, School of Civil and Environmental Engineering, Ningbo University, Ningbo, Zhejiang 315211, PR China

ARTICLE INFO

Keywords:
Rock discontinuities
Clustering analysis
Optimization
Neutrosophic genetic algorithm
Soft computing

ABSTRACT

This paper presents the neutrosophic genetic algorithm (NGA) to address the research gap in the application of neutrosophic theory in conjunction with genetic algorithms. NGA introduces three distinct solution spaces—truth, falsity, and indeterminacy—enabling it to entirely encompass neutrosophic solution spaces in the operational process. Fine-tuning in the true solution space (TSS), adaptive regeneration in the false solution space (FSS), and modified crossover and mutation operations in the indeterminate solution space (ISS) enhance NGA ability to navigate away from local optima while reducing computational complexity. Evaluation against several prior algorithms based on the CEC2017 test suites demonstrates the superior performance of NGA, achieving the highest overall score of 92.11% in various problems and conditions. Sensitivity analysis of NGA parameters provides significant insights into algorithm performance variations, emphasizing the substantial impact of these parameters on the NGA's performance. The application of NGA to optimize the K-means method for clustering analysis of rock discontinuity sets showcases its efficiency and potential for practical applications in related fields, highlighting its advantages over other methods. This research establishes NGA as an innovative and efficient approach to address imprecision, incompleteness, and uncertainty in practical data scenarios, with significant implications for future development and applications.

1. Introduction

The field of multimodal and high-dimensional optimization is vital in computer science and engineering, and researchers have developed various soft computing strategies, such as simulated annealing (SA), genetic algorithm (GA), immune algorithm (IA), particle swarm optimization (PSO), the particle swarm optimization algorithm based on multiple adaptive strategies (MAPSO), ant colony optimization algorithm (ACO), and state transition algorithm (STA). Among these, GA, inspired by natural selection, proves to be a parallel, efficient, and robust search and optimization technique, particularly suitable for solving large optimization problems. Notably, the concept of using real-valued coding for the representation of chromosomes in GA has been developed and expanded upon by various researchers in the field of evolutionary computation. This has led to the development of the real-coded genetic algorithm (RCGA) and its widespread application in real-world scenarios (Almabsout et al., 2023; Li et al., 2023; Moorthy & Marappan, 2022; Narang & De, 2023; Wei & Cheng, 2022).

In practical applications, data often presents imprecision, incompleteness, and uncertainty. Neutrosophic theory, introduced by Smarandache (1998), has emerged as a framework for addressing incomplete and indeterminate information, extensively applied in various fields like image processing, decision-making, medical diagnosis, and fault diagnosis. Researchers have endeavored to combine neutrosophic theory with Genetic Algorithms (GA) to effectively handle optimization problems with indeterminate information. For instance, Kumar et al. (2015) developed a hybrid model of fuzzy cognitive maps and GA to handle decision-making scenarios characterized by unclear distinctions. Ashour et al. (2018) implemented a novel skin lesion detection method based on GA, optimizing the neutrosophic set (NS) operation to reduce indeterminacy in dermoscopy images. Elwahsh et al. (2018) proposed an approach utilizing a neutrosophic intelligent system based on GA for classifying security attacks in mobile ad hoc networks. Ye (2019) presented a method employing a cosine similarity measure of single-valued neutrosophic sets (SVNSs) and GA to improve the existing proportional-integral-derivative (PID) tuning method. Recently, Zhang et al. (2023) proposed a novel concept called "neutrosophic adaptive clustering optimization". In their study, they considered the crossover

* Corresponding author.
E-mail addresses: yongrui@nbu.edu.cn (R. Yong), 2211110037@nbu.edu.cn (H. Wang), yejun1@nbu.edu.cn (J. Ye), dushigui@nbu.edu.cn (S. Du).

https://doi.org/10.1016/j.eswa.2023.122973
Received 20 September 2023; Received in revised form 14 December 2023; Accepted 14 December 2023
Available online 29 December 2023

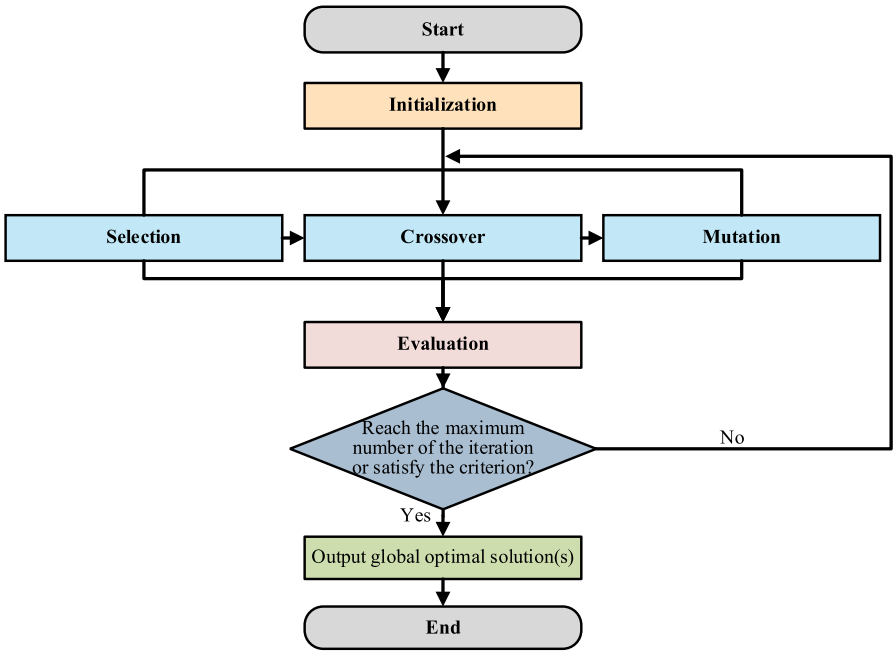


Fig. 1. General flowchart of RCGA.

effect as NS and formed their neutrosophic fitness function in GA. Neutrosophic theory offers a suitable framework for dealing the problems with incomplete and indeterminate information, but its application in conjunction with genetic algorithms has primarily focused on comprehensive usage, neglecting specific enhancements to the GA algorithm itself. A specific neutrosophic genetic algorithm (NGA), that can entirely encompass the neutrosophic solution spaces in the operational process, has not been developed to date, presenting a significant research gap.

To fundamentally improve RCGA based on neutrosophic theory, this paper aims to propose NGA and apply it to optimization problems. NGA has several innovative aspects and advantages. One key aspect of NGA is its unique approach to solution space division, where each generated population is categorized into three distinct solution spaces: the true solution space, the false solution space, and the indeterminate solution space. This nuanced categorization enables a more comprehensive exploration of the solution landscape, enhancing the algorithmic capability to address complex optimization challenges. Additionally, NGA represents solutions as real-valued vectors, in contrast to the binary string representation used in conventional genetic algorithms. This approach brings several benefits, including higher precision, elimination of coding and decoding procedures, efficient exploration of large solution spaces, and simplified computation. Moreover, NGA demonstrates higher convergence efficiency with fewer generations and increased accuracy compared to traditional genetic algorithm methods, showcasing reduced sensitivity to the initial population size and contributing to robust performance. Furthermore, NGA ability to identify one or more optimal and suboptimal solutions provides a range of valuable options for engineering applications, particularly in scenarios where access to the optimal solutions is limited. Thus, NGA is a promising algorithm with the potential to tackle diverse and challenging optimization problems across various domains.

Rock discontinuities significantly impact the mechanical and hydraulic properties of rock masses (Huang et al., 2019; Wu & Kulatilake, 2012). Given the protracted and unpredictable nature of geological processes, the presence of uncertainty and variability in these discontinuities is inevitable (Du et al., 2022). Clustering analysis has been proven to be an effective tool to categorize random discontinuities into distinct sets within a specific geological range. The clustering analysis was initially employed for identifying discontinuity sets by Shanley & Mahtab (1976), and since then various methods such as fuzzy C-means (FCM) clustering (Hammah & Curran, 1998; Song et al., 2017), K-means clustering (Tokhmechi et al., 2011; Zhou & Maerz, 2002), spectrum clustering (Jimenez, 2008; Jimenez-Rodriguez & Sitar, 2006), and affinity propagation (AP) algorithm (Liu et al., 2017) have been extensively applied for statistical analysis of discontinuities. However, these methods exhibit limitations including computational complexity, sensitivity to initial parameters, and susceptibility to noise and outliers, indicating the necessity for further exploration and study of rock discontinuity set clustering analysis. Considering GA advantages in solving optimization problems, many researchers proposed GA-based clustering algorithms to find optimal solutions for grouping rock discontinuities (Cui & Yan, 2020; Kemeny & Post, 2003; Lu et al., 2007). However, these clustering methods have their limitations. Firstly, they may exhibit slow convergence due to their stochastic and population-based search nature. Secondly, the choice of coding schemes in GAs can significantly impact the algorithmic performance and convergence. Furthermore, GAs are prone to getting trapped in local optima. This study aims to propose NGA, trying to fundamentally improve the RCGA based on neutrosophic theory, and apply NGA to the clustering analysis of rock discontinuity sets.

This paper is constructed in the following parts. Section 2 proposes the NGA approach based on a neutrosophic extension of RCGA. Section 3 validates the effectiveness of NGA through numerical examples. Section 4 applies the NGA approach to the clustering of rock discontinuity sets. Section 5 presents a case study on the clustering analysis of rock discontinuity sets and compares the proposed NGA approach with existing methods to validate the effectiveness and superiority of the NGA approach. Finally, Section 6 concludes the paper and discusses potential areas for future work.

2. Neutrosophic genetic algorithm

GA operates on a population of individual solutions, which are represented as chromosomes or strings. These chromosomes are evaluated based on a fitness value, and stochastic genetic operators, such as mutation, crossover, and selection, guide the evolution of the population through generations. The algorithm terminates when a specific

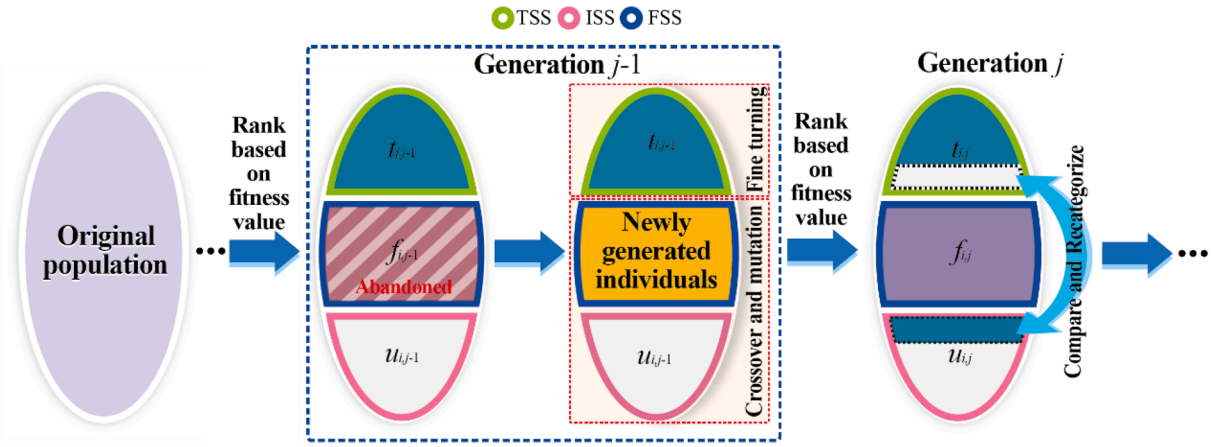


Fig. 2. Schematic diagram of NGA.

termination criterion is reached.

In GA, binary encoding is commonly employed. However, this method has limitations, including high computational complexity, inadequate accuracy for practical engineering applications, and the need for decoding, leading to limited control over the process. To address these issues, Michalewicz et al. (1992) introduced RCGA, in which solutions are represented as real-valued vectors instead of binary strings. It is particularly effective in solving optimization problems involving continuous variables, such as mathematical functions and engineering design. The general flowchart of RCGA is shown in Fig. 1. Researchers have focused on improving RCGA performance by optimizing parameters, introducing new mutation schemes and selection methods, exploring parallel processing techniques, and hybridizing with other optimization algorithms to enhance efficiency.

According to the neutrosophic concept introduced by Smarandache (1998), there is true, false, and indeterminate information in real-world problems. Applying the neutrosophic concept in RCGA, NGA is presented for the first time. As shown in Fig. 2, it illustrates the main components in NGA. The fitness values of all individuals in the population are calculated using the objective function. Then, we can order the individuals based on their fitness values. Each generated population subject to their corresponding fitness values is divided into three solution spaces: the true solution space (TSS), the false solution space (FSS), and the indeterminate solution space (ISS) (Fig. 2). The TSS contains the individuals that evolve toward high fitness values, reflecting the true information. As the evolution proceeds, the fitness values of these individuals increase continuously. Conversely, the FSS comprises individuals with low fitness values. Then, the ISS includes the individuals whose evolutionary direction is uncertain and cannot be definitively categorized as either high or low fitness values. Some of these individuals may evolve toward a high fitness value or may evolve toward a low fitness value during the evolution process.

Based on the arithmetic encoding method, the individual is initiated by the following equation:

$$P_{i,0} = lb + \delta_i \cdot (ub - lb) \tag{1}$$

where $P_{i,0}$ ($i = 1, 2, \ldots, n$) is the individual in the initial population, δ_i is a random value from 0 to 1, n is the population size, and lb and ub are the lower and upper limits of the population.

In the TSS, the individuals are denoted as $t_{i,j-1}$ ($i = 1, 2, \ldots, p$) in the j-1th iteration, where p represents the size of TSS, which depends on the number of optimal and suboptimal solutions. These individuals are sorted in descending order based on their fitness values. The first individual represents the individual with the highest fitness value in the current iteration. In the TSS, the individual with the highest fitness value in the final iteration is the optimal solution, and the rest are suboptimal solutions.

To access an improvement in the search efficiency of the optimal and suboptimal solutions, as depicted in Fig. 2, each of the remaining individuals $t_{i,j}$ ($i = 2, 3, \ldots, p$) undergoes a fine-tuning process to enhance their performance according to the equation:

$$t_{i,j} = \frac{\left[10^d \cdot t_{i,j-1}\right] + r}{10^d} \tag{2}$$

where d is an accuracy control parameter used for controlling the number of decimal places, r is a random variable in $[-1, 1]$, and $[\cdot]$ here is the round operator. According to Eq. (2), the fine-tuning process can introduce minor adjustments to decimal places through the utilization of a flexible accuracy control parameter.

In the FSS, the individuals are denoted as $f_{i,j-1}$ ($i = 1, 2, \ldots, q$) in the j-1th iteration, where q represents the size of FSS, which depends on the number of lower-fitted individuals that need to be abandoned (Fig. 2). Roulette wheel selection (RWS), introduced by Holland (1992), is a widely used selection strategy in evolutionary algorithms due to its simplicity and ease of implementation. However, in some multimodal problems, its potential limitation may be exposed as individuals with slightly higher fitness values may dominate the selection process, reducing genetic diversity and potentially leading the population to fall into traps. Thus, we tried to give the lower-fitted individuals in the FSS a more aggressive change according to the neutrosophic methodology. The individuals in the FSS were abandoned. To preserve the same population size and diversity, a group of new individuals with the same number as the ones abandoned was generated using Eq. (1). As shown in Fig. 2, these newly generated individuals are then allocated to the ISS, where they actively participate in subsequent evolutionary processes. It should be noted that in each generation, there is no need to calculate the fitness value of the newly generated individuals, which ensures a low computational complexity. This approach refreshes the population, and offers opportunities for further exploration and improvement, preventing the population from falling into the trap of local optimal values.

In the ISS, the individuals are denoted as $u_{i,j-1}$ ($i = 1, 2, \ldots, n - p$) in the j-1th iteration. As shown in Fig. 2, the crossover and mutation operators are carried out in the ISS to generate new individuals. The crossover operation is based on merging genetic material from multiple individuals to generate offspring with novel combinations of traits. The mutation operation involves randomly changing one or more genes in a chromosome gene. These two operations facilitate the exploration of various genetic combinations and introduce variation into the population, thus enhancing the chances of discovering optimal solutions.

Each individual in the ISS has an exactly equal chance of undergoing the crossover process. Crossover points in the population are randomly determined to introduce variability. To do so, two different parameters

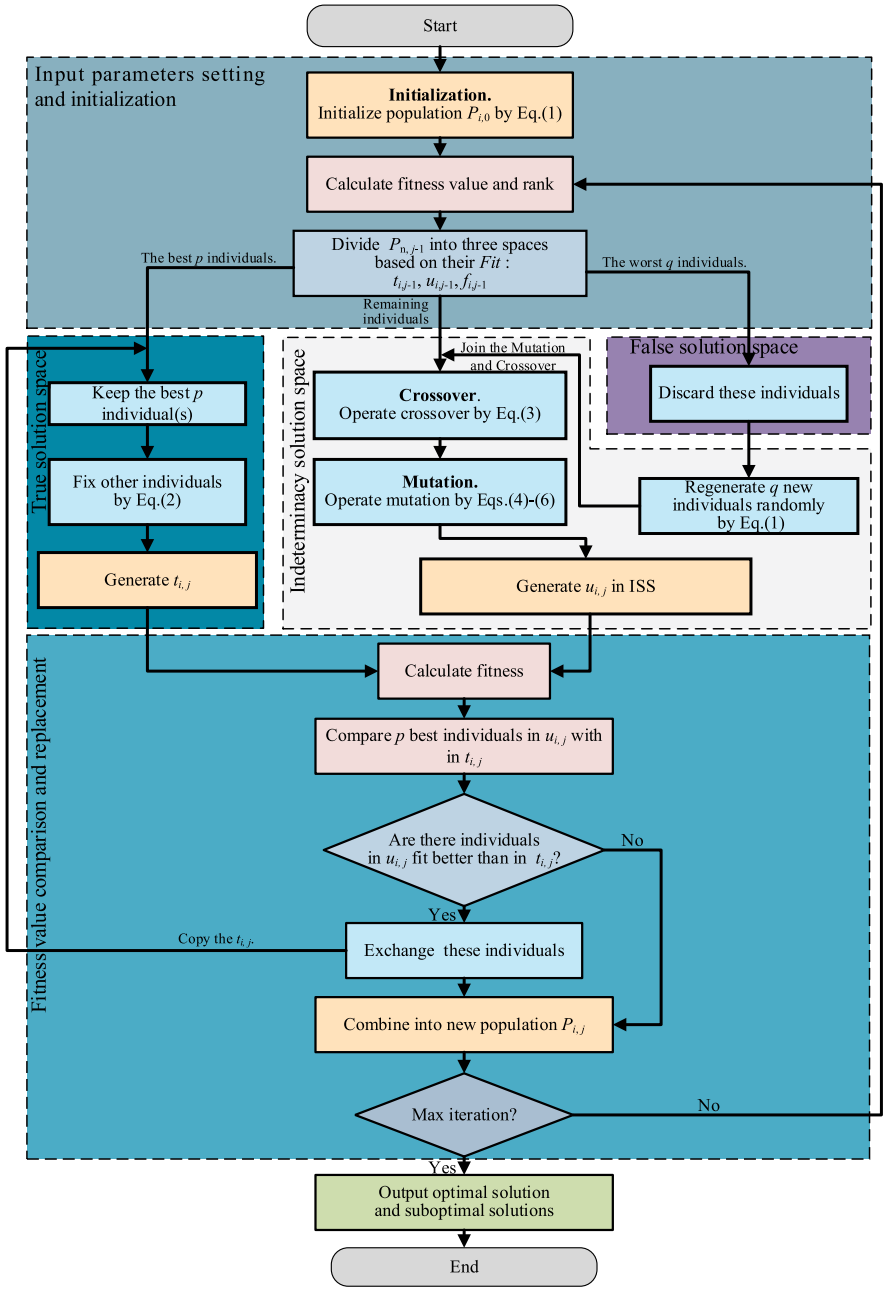


Fig. 3. Flowchart of the proposed NGA.

that obey a uniform distribution are utilized to control the crossover points. This random selection ensures that crossover occurs at different positions equally throughout the population, promoting diversity and exploration of new genetic combinations. Based on the arithmetical crossover operators for RCGA by Michalewicz et al. (1992), a modified crossover operator was introduced in NGA. The input parameters for the crossover operator are expressed as follows:

$$\begin{cases} n \in N, \\ u_{i,j-1} \in R, \\ P_c \in [0,1], \\ p, q \in N. \end{cases}$$

where N represents the set of non-negative integers, R represents the set of real numbers, and P_c is the possibility of crossover.

The probabilistic selection strategy is performed here for the crossover operation. The decision to perform crossover is determined by comparing it with a certain probability value, typically using a random number (Holland, 1992). A random variable P_1 is used to determine the specific positions in the crossover process, which is drawn from a uniform distribution ranging from 0 to 1. If $P_1 > P_c$, the crossover process is operated as

$$\begin{cases} u_{a,j} = P_1 \cdot u_{a,j-1} + (1 - P_1) \cdot u_{b,j-1} \\ u_{b,j} = P_1 \cdot u_{b,j-1} + (1 - P_1) \cdot u_{a,j-1} \end{cases}, \quad i \neq k \tag{3}$$

where $u_{a,j\text{-}1}$ and $u_{b,j\text{-}1}$ are two individuals in the ISS; $u_{a,j}$ and $u_{b,j}$ are the offspring of $u_{a,j\text{-}1}$ and $u_{b,j\text{-}1}$.

Through the crossover operation, it combines beneficial traits from different parent solutions to produce potentially superior offspring. It emphasizes the combination of existing genetic material to produce offspring. This process enables the propagation of promising genetic material and helps to explore the solution space more effectively, potentially leading to better solutions over successive generations.

Table 1
General pseudocode of the NGA.

Algorithm: **NGA**
INPUT: Population size n; Maximum number of iterations Max; Possibility of crossover P_c; Possibility of mutation P_m; Size of TSS p; Size of FSS q.
OUTPUT: Global optimal solution; Suboptimal solutions.
Begin
Initiate population using Eq. (1), build the initial population $P_{i,0}$ ($i = 1, 2, \ldots, n$).
Calculate the fitness value of each individual.
Set iteration counter $j = 1$.
While ($j < Max$)
Select the best p individuals to add to the TSS, build $t_{i,j-1}$ ($i = 1, 2, \ldots, p$).
Fine-tune $t_{i,j-1}$ ($i = 1, 2, \ldots, p$) referring to Eq. (2).
Select the worst q individuals to add to the FSS, and discard these individuals.
Regenerate q individuals referring to Eq. (1), denoted by $f_{i,j}$ ($i = 1, 2, \ldots, q$).
Combine the remaining individuals and $f_{i,j}$ ($i = 1, 2, \ldots, q$), build $u_{i,j-1}$ ($i = 1, 2, \ldots, n - p$) in the ISS.
If $P_1 > P_c$,
Operate crossover based on Eq. (3).
End if
If $P_2 > P_m$,
Operate mutation based on Eqs. (4)–(6).
End if
Output the operated individuals in the ISS, denoted by $u_{i,j}$ ($i = 1, 2, \ldots, n - p$).
Calculate the fitness value of each individual.
Compare the fitness values of the best p individuals in $u_{i,j}$ ($i = 1, 2, \ldots, n - p$) with $t_{i,j}$ ($i = 1, 2, \ldots, p$).
If $u_{i,j} > t_{i,j}$,
Replace individuals in $t_{i,j}$ and $u_{i,j}$.
End if
Combine $t_{i,j}$ and $u_{i,j}$ into a new population, denoted as $P_{i,j}$ ($i = 1, 2, \ldots, n$).
Increment the counter by 1.
End while
Return the optimal solution and suboptimal solutions.
End

The input parameters for the mutation operation are expressed as follows:

$$\begin{cases} n \in N, \\ u_{i,j-1} \in R, \\ P_m \in [0,1], \\ p, q \in N. \end{cases}$$

where P_m is the possibility of mutation.

The probabilistic selection strategy is also performed for the mutation operation. A random variable P_2 is utilized to initiate the mutation process, which is drawn from a uniform distribution spanning from 0 to 1. If $P_2 > P_m$, the following mutation operation steps are taken to generate new individuals.

Firstly, the decimal part of the selected individual is extracted by the following equation:

$$\alpha_{i,j-1} = u_{i,j-1} - \left[u_{i,j-1}\right], \tag{4}$$

where $u_{i,j}$ is the offspring of the selected individual $u_{i,j-1}$,$\alpha_{i,j-1}$($i = 1, 2, \ldots, n - p$) is the decimal part of the selected individual, and $[\cdot]$ is the round operator.

Next, a random integer part is regenerated within the predefined boundaries as follows:

$$A_{i,j} = [lb + \delta_i \cdot (ub - lb)] \tag{5}$$

where $A_{i,j}$ ($i = 1, 2, \ldots, n - p$) is the integer part prepared for the offspring.

Finally, a new individual is created by adding $A_{i,j}$ and $\alpha_{i,j-1}$:

$$u_{i,j} = A_{i,j} + \alpha_{i,j-1} \tag{6}$$

Through the mutation operation, it introduces new genetic materials into the population that were not previously presented and can sometimes cause fundamental changes in the individuals. It focuses on the introduction of novel genetic material through random changes and maintains genetic diversity.

Table 2
Parameter settings of the eight algorithms.

Name	Parameter settings
MAPSO	Group number = 10*D, Group size = 3, $R_1 = 10$, $R_2 = 100$.
IA	Group number = 10*D, $P_0 = 0.7$, $\alpha = 1$, $\beta = 1$, $\delta = 0.2$, Clone = 10.
STA	Group number = 10*D, $\alpha_{max} = 1$, $\alpha_{min} =$ 1e-4, $\beta = 1$, $\gamma = 1$, $\delta = 1$, SE = 30, fc = 2.
PSO	Group number = 10*D, $c_1 = 1.5$, $c_2 = 1.5$, $W_{max} = 0.8$, $W_{min} = 0.4$.
SA	Group number = 10*D, $K = 0.998$, $S = 0.01$, $T = 100$, $YZ =$ 1e-8, $P = 0$.
ACOR	Group number = 10*D, N_{sample} = Group number *4, $q = 0.8$, $\xi = 1$.
RCGA	Group number = 10*D, Selection type= 'Roulette Wheel', $P_m = 0.4$, $P_c =$ 0.15.
NGA	Group number = 10*D, Selection type= 'Random', $P_c = 0.15$, $P_m = 0.4$, $p = 1$, $q = 8$.

After performing crossover and mutation operations, operations in the ISS introduce more diversity, random generate, crossover, and mutation may produce some better individuals than the individuals in the TSS. These individuals should be categorized into the TSS and as a leading position for the next step of iteration. Thus, the comparison and recategorizing process are made based on the fitness value of the individuals. Let $u_{i,j}$ ($i = 1, 2, \ldots, p$) be the first p best fitness value in the ISS, $t_{i,j}$ ($i = 1, 2, \ldots, p$) be individuals in the TSS. If any $u_{i,j}$ in the ISS has a higher fitness value than $t_{i,j}$, then this individual should be recategorized into the TSS, while the individual $t_{i,j}$ in the TSS with a lower fitness value should be correspondingly recategorized into the ISS (Fig. 2). The comparison and recategorizing process let the population have more chance to jump out of local optima, then the individuals who have the best fitness value always stay in the TSS. Additionally, this strategy reduces the need for the whole population to sort by order, only 2*p individuals will be sorted by order, which will reduce the computational complexity from $O(n*\log(n))$ to $O(n)$.

Fig. 3 shows the flowchart of the proposed NGA. The general pseudocode of the proposed NGA is shown in Table 1.

3. Experimental verification and comparisons

3.1. Experiment setup

Statistical comparisons are crucial for obtaining reliable conclusions of intelligence algorithms (Carrasco et al., 2020). Thus, to verify how NGA performs in different problems and circumstances, CEC2017 test suites (Awad et al., 2017) are utilized in this study. The test suites include two unimodal (f_1 and f_3), seven multimodal (f_4 - f_{10}), 10 hybrid (f_{11} - f_{20}), and 10 composition functions (f_{21} - f_{30}). NGA is compared to prior algorithms including MAPSO (Wei et al., 2020), IA (Farmer et al., 1986), STA (Zhou et al., 2022), PSO (Bonyadi & Michalewicz, 2017), SA (Kirkpatrick et al., 1983), ACOR (Zhou et al., 2023), and RCGA (Song et al., 2022). The comparison is performed based on a dimension case (e.g. $D = 10$) and the parameter configurations utilized for different algorithms are tabulated in Table 2.

For the termination conditions, the maximum number of function evaluations (MaxFEs) is set to 20, 000 * D for all dimension cases, all algorithms will be stopped when reaching MaxFEs. For the convergence criteria, if the distance between two iteration results is smaller than the minimum precision 1e-8, then the algorithm should be stopped. The parameter settings for NGA are identical to those of RCGA, facilitating an easier comparison, as NGA is an enhanced algorithm derived from RCGA. The following testing experiments were conducted on a Windows 10 system running MATLAB 2022b, equipped with an Intel(R) Xeon(R) E5-2666 v3@2.9 GHz CPU and 64.0 GB of RAM.

3.2. Comparison based on the CEC2017 test suites

The performance of each algorithm is evaluated based on two key

Table 3
Comparison results of solution accuracy on the CEC2017 test suites ($D = 10$).

		MAPSO	IA	PSOTA	PSO	SA	ACOR	RCGA	NGA
f_1	Mean	4.95E + 03	4.32E + 03	5.92E + 03	3.69E + 02	3.53E + 05	2.35E + 03	1.34E + 04	2.78E + 04
	SD	1.19E + 04	9.64E + 02	4.06E + 03	7.94E + 02	8.77E + 04	3.20E + 03	2.31E + 04	5.73E + 04
f_2	Mean	3.30E + 00	3.62E + 01	1.18E + 03	1.99E-02	4.50E + 01	8.75E-05	4.35E + 05	3.53E + 05
	SD	2.32E + 01	6.99E + 01	2.40E + 03	6.66E-02	4.65E + 01	4.52E-05	1.76E + 06	1.41E + 06
f_3	Mean	1.09E + 01	6.01E + 02	5.67E-02	8.87E-14	3.86E + 01	6.95E-07	2.48E + 03	1.34E + 03
	SD	1.78E + 01	5.40E + 02	3.86E-01	1.27E-13	1.18E + 01	3.69E-06	1.36E + 03	7.97E + 02
f_4	Mean	2.44E + 00	1.82E-01	1.04E + 01	5.98E-01	5.27E + 01	3.14E + 00	7.05E + 00	7.01E + 00
	SD	4.44E-01	4.61E-01	1.56E + 01	3.80E + 00	2.49E + 01	7.03E-02	6.54E-01	4.77E-01
f_5	Mean	2.25E + 01	1.70E + 01	2.72E + 01	2.55E + 01	1.33E + 02	2.52E + 01	9.76E + 00	1.68E + 01
	SD	9.31E + 00	5.14E + 00	1.15E + 01	8.86E + 00	5.05E + 01	3.93E + 00	3.71E + 00	6.06E + 00
f_6	Mean	8.81E-05	1.50E + 01	2.36E + 00	1.71E + 00	7.37E + 01	0.00E + 00	6.30E-03	1.41E-02
	SD	3.18E-04	4.58E + 00	4.01E + 00	1.44E + 00	1.79E + 01	0.00E + 00	1.66E-03	4.22E-03
f_7	Mean	4.63E + 01	5.40E + 01	4.76E + 01	2.80E + 01	1.60E + 02	3.56E + 01	2.72E + 01	3.60E + 01
	SD	1.13E + 01	1.37E + 01	1.59E + 01	9.26E + 00	5.77E + 01	3.24E + 00	5.02E + 00	4.43E + 00
f_8	Mean	2.31E + 01	1.61E + 01	2.61E + 01	2.07E + 01	1.09E + 02	2.51E + 01	9.10E + 00	1.27E + 01
	SD	9.48E + 00	4.36E + 00	1.13E + 01	7.94E + 00	2.86E + 01	3.73E + 00	4.10E + 00	5.89E + 00
f_9	Mean	1.79E-03	2.20E + 02	1.05E + 02	2.73E + 00	3.94E + 03	0.00E + 00	8.09E-03	2.07E-02
	SD	1.27E-02	8.86E + 01	1.35E + 02	4.71E + 00	2.20E + 03	0.00E + 00	1.24E-02	2.43E-02
f_{10}	Mean	3.69E + 02	4.65E + 02	7.48E + 02	7.25E + 02	1.09E + 03	8.72E + 02	6.23E + 02	5.42E + 02
	SD	3.83E + 02	1.51E + 02	3.19E + 02	2.50E + 02	3.14E + 02	2.54E + 02	2.51E + 02	2.82E + 02
f_{11}	Mean	3.31E + 00	1.27E + 01	2.83E + 01	2.42E + 01	7.47E + 01	5.68E + 00	6.11E + 00	7.97E + 00
	SD	1.96E + 00	6.04E + 00	2.57E + 01	1.35E + 01	4.12E + 01	7.84E-01	2.18E + 00	2.62E + 00
f_{12}	Mean	7.20E + 04	1.33E + 04	4.97E + 05	1.87E + 04	3.66E + 05	8.61E + 03	1.27E + 05	1.38E + 05
	SD	2.08E + 05	1.09E + 04	5.65E + 05	5.92E + 04	2.94E + 05	4.51E + 03	2.03E + 05	1.65E + 05
f_{13}	Mean	2.39E + 03	4.94E + 02	7.76E + 03	2.06E + 02	1.52E + 04	5.94E + 03	6.11E + 03	5.81E + 03
	SD	3.77E + 03	1.75E + 02	7.51E + 03	3.37E + 02	1.19E + 04	4.59E + 03	4.61E + 03	4.94E + 03
f_{14}	Mean	3.49E + 01	5.82E + 01	5.62E + 02	4.50E + 01	1.46E + 03	2.18E + 02	2.52E + 02	2.04E + 02
	SD	2.34E + 01	1.91E + 01	1.06E + 03	2.06E + 01	1.71E + 03	4.69E + 02	4.50E + 02	3.99E + 02
f_{15}	Mean	5.41E + 01	3.32E + 02	3.76E + 02	2.99E + 01	7.36E + 03	4.72E + 02	8.87E + 02	7.97E + 02
	SD	4.50E + 01	2.15E + 02	8.11E + 02	2.84E + 01	5.57E + 03	8.20E + 02	1.36E + 03	1.20E + 03
f_{16}	Mean	8.56E + 00	3.73E + 01	1.97E + 02	1.41E + 02	4.03E + 02	1.19E + 01	2.12E + 01	1.96E + 01
	SD	2.85E + 01	3.67E + 01	1.33E + 02	1.23E + 02	1.48E + 02	1.57E-01	3.34E + 01	3.67E + 01
f_{17}	Mean	2.16E + 01	3.08E + 01	4.76E + 01	5.70E + 01	2.11E + 02	4.47E + 01	2.22E + 01	2.93E + 01
	SD	1.84E + 01	6.16E + 00	4.17E + 01	3.79E + 01	1.32E + 02	1.97E + 00	1.72E + 01	1.19E + 01
f_{18}	Mean	1.32E + 04	6.18E + 02	1.20E + 04	1.51E + 02	1.68E + 04	2.56E + 04	7.49E + 03	6.50E + 03
	SD	1.42E + 04	4.50E + 02	9.52E + 03	7.78E + 02	1.52E + 04	1.34E + 04	6.56E + 03	5.80E + 03
f_{19}	Mean	5.11E + 01	5.87E + 01	3.69E + 03	1.26E + 01	3.29E + 03	5.41E + 03	7.39E + 02	1.69E + 03
	SD	3.82E + 01	4.53E + 01	4.69E + 03	1.12E + 01	3.45E + 03	7.35E + 03	7.86E + 02	2.45E + 03
f_{20}	Mean	6.64E + 00	4.40E + 01	3.33E + 01	4.07E + 01	3.68E + 02	2.56E-01	6.27E + 00	7.35E + 00
	SD	1.17E + 01	9.08E + 00	3.46E + 01	3.53E + 01	1.16E + 02	1.86E-01	6.86E + 00	8.80E + 00
f_{21}	Mean	1.45E + 02	9.85E + 01	1.18E + 02	1.99E + 02	3.02E + 02	2.24E + 02	1.50E + 02	1.27E + 02
	SD	5.84E + 01	2.04E + 01	4.22E + 01	5.38E + 01	6.46E + 01	1.35E + 01	4.84E + 01	4.72E + 01
f_{22}	Mean	1.01E + 02	6.41E + 01	9.80E + 01	9.72E + 01	1.05E + 03	1.04E + 02	1.01E + 02	9.96E + 01
	SD	1.48E + 01	2.36E + 01	3.23E + 01	2.10E + 01	5.58E + 02	1.21E + 00	1.40E + 00	1.73E + 01
f_{23}	Mean	3.13E + 02	3.11E + 02	3.38E + 02	3.37E + 02	5.04E + 02	3.20E + 02	3.14E + 02	3.13E + 02
	SD	8.04E + 00	6.65E + 01	1.28E + 01	1.92E + 01	1.94E + 02	3.83E + 00	5.20E + 00	6.01E + 00
f_{24}	Mean	3.28E + 02	1.06E + 02	3.06E + 02	3.38E + 02	4.37E + 02	3.54E + 02	3.19E + 02	2.83E + 02
	SD	6.82E + 01	4.08E + 01	1.18E + 02	8.79E + 01	8.95E + 01	3.06E + 00	6.64E + 01	1.02E + 02
f_{25}	Mean	4.11E + 02	1.73E + 02	4.30E + 02	4.20E + 02	4.86E + 02	4.44E + 02	4.41E + 02	4.39E + 02
	SD	2.12E + 01	1.16E + 02	2.86E + 01	5.15E + 01	3.55E + 01	1.17E + 01	1.30E + 01	1.53E + 01
f_{26}	Mean	3.00E + 02	1.19E + 02	4.67E + 02	3.15E + 02	8.57E + 02	5.27E + 02	3.13E + 02	3.28E + 02
	SD	2.22E-07	9.60E + 01	9.84E + 01	1.55E + 02	4.41E + 02	3.86E + 02	4.76E + 01	2.33E + 01
f_{27}	Mean	3.93E + 02	4.09E + 02	4.04E + 02	4.14E + 02	5.17E + 02	3.94E + 02	3.95E + 02	3.96E + 02
	SD	1.47E + 00	7.03E + 00	1.68E + 01	2.45E + 01	2.71E + 02	4.79E-01	1.69E + 00	2.02E + 00
f_{28}	Mean	3.97E + 02	2.05E + 02	4.41E + 02	4.64E + 02	6.34E + 02	6.11E + 02	4.50E + 02	3.94E + 02
	SD	1.40E + 02	1.48E + 02	9.74E + 01	1.63E + 02	1.50E + 02	6.49E + 00	1.08E + 02	1.17E + 02
f_{29}	Mean	2.72E + 02	2.76E + 02	3.08E + 02	3.18E + 02	4.48E + 02	2.60E + 02	2.77E + 02	2.94E + 02
	SD	1.40E + 01	1.73E + 01	5.08E + 01	5.17E + 01	1.10E + 02	7.43E + 00	1.66E + 01	1.62E + 01
f_{30}	Mean	1.70E + 05	1.08E + 04	3.71E + 05	7.16E + 05	2.78E + 04	3.24E + 05	9.05E + 04	6.86E + 04
	SD	3.75E + 05	1.12E + 04	5.19E + 05	1.08E + 06	3.18E + 04	4.33E + 05	1.76E + 05	1.86E + 05

aspects: solution accuracy and computational efficiency. The assessment of solution accuracy is measured in terms of mean value (Mean) and standard deviation (SD). Assuming equal weight values for these aspects, the weight value for the score based on solution accuracy is set at 50 %, while the weight value for the score based on computational efficiency is also set at 50 %.

To quantify the solution accuracy, we ranked the algorithms based on their mean value. For each function f_k ($k = 1, 2, \ldots, 30$), the score of solution accuracy S_k ranges from 1 to 8 according to the following criteria: The algorithm yielding the minimum value is assigned the highest score of 8, while conversely, the algorithm yielding the maximum value is assigned the lowest score of 1. We can evaluate each algorithm using *Score*1 by summing all the obtained S_k based on different functions.

To assess computational efficiency, the algorithms were ranked based on their time usage. The algorithms were ranked according to their time usage. For each function f_k ($k = 1, 2, \ldots, 30$), the computational efficiency score C_k ranges from 1 to 8 based on the following criteria: The algorithm with the minimum time usage is assigned the highest score of 8, while the algorithm with the maximum time usage is assigned the lowest score of 1. We can evaluate each algorithm using *Score*2 by summing all the obtained C_k based on different functions.

Table 4
Comparison of time usage on the CEC2017 test suites f_1-f_{30} ($D = 10$) (in seconds).

	MAPSO	IA	PSOTA	PSO	SA	ACOR	RCGA15	NGA
f_1	1.2744	1.3558	0.0791	0.6632	0.5529	8.2215	0.5871	0.3388
f_2	1.2684	1.8317	0.0699	0.6747	0.5655	7.2162	0.5946	0.3351
f_3	1.3343	2.5569	0.0836	0.6589	0.5754	2.6586	0.5919	0.3367
f_4	1.2072	2.5212	0.0409	0.6121	0.5410	2.5734	0.5860	0.3287
f_5	1.3439	1.1077	0.0477	0.6495	0.5564	2.6222	0.5922	0.3291
f_6	1.3836	1.0862	0.1984	0.7802	0.6807	2.7199	0.6150	0.3532
f_7	1.3382	0.9442	0.0710	0.6715	0.5798	2.5984	0.5964	0.3400
f_8	1.3373	0.9711	0.0497	0.6545	0.5591	2.5696	0.5900	0.3321
f_9	1.3092	1.0612	0.1472	0.7390	0.6296	2.6386	0.6256	0.3489
f_{10}	1.3849	1.0345	0.1135	0.6897	0.6096	2.6691	0.6122	0.3424
f_{11}	1.2620	0.9764	0.0624	0.6349	0.5631	2.6336	0.6039	0.3367
f_{12}	1.3389	0.9925	0.0809	0.6910	0.5396	2.6229	0.6118	0.3403
f_{13}	1.3661	0.9806	0.0811	0.6970	0.5638	2.6128	0.6077	0.3406
f_{14}	1.4084	0.9380	0.0754	0.6847	0.5781	2.6365	0.6129	0.3396
f_{15}	1.3956	0.9564	0.0657	0.6704	0.5573	2.5973	0.6153	0.3370
f_{16}	1.3665	0.9666	0.0826	0.6854	0.5570	2.6278	0.6188	0.3414
f_{17}	1.4969	1.0927	0.2110	0.7742	0.6831	2.7814	0.6419	0.3640
f_{18}	1.3960	0.9503	0.0757	0.6578	0.5699	2.6236	0.6180	0.3382
f_{19}	1.6697	1.3382	0.5211	1.0534	0.9293	3.0462	0.6891	0.3973
f_{20}	1.5069	1.1094	0.2281	0.7844	0.7263	2.7347	0.7169	0.3648
f_{21}	1.4570	1.0486	0.2353	0.8021	0.7153	2.7690	0.6675	0.3641
f_{22}	1.4442	1.1778	0.2960	0.8706	0.7758	2.8496	0.7391	0.3752
f_{23}	1.5717	1.1890	0.3502	0.8824	0.7772	2.9197	0.6923	0.3864
f_{24}	1.5671	1.1999	0.2929	0.8764	0.7912	2.8152	0.6805	0.3790
f_{25}	1.4625	1.1577	0.3294	0.8973	0.8076	2.8299	0.6906	0.3796
f_{26}	1.5044	1.3064	0.4092	0.9467	0.8401	2.9369	0.7076	0.3966
f_{27}	1.7137	1.3267	0.5048	1.0766	0.9556	2.9911	0.7265	0.4072
f_{28}	1.5542	1.3037	0.4367	0.9955	0.8609	2.8830	0.7092	0.3934
f_{29}	1.6974	1.2537	0.4654	1.0068	0.8711	2.9771	0.7212	0.4025
f_{30}	1.8263	1.5570	0.7524	1.2641	1.1244	3.1938	0.7768	0.4343

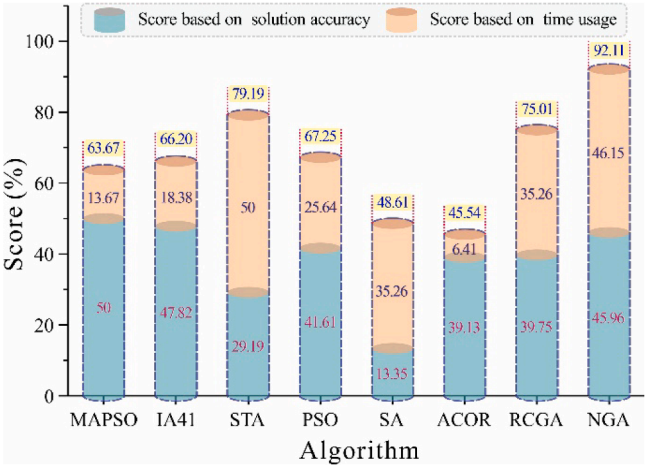


Fig. 4. Comparison of the scores of different algorithms.

The overall score of each algorithm *SR* can be normalized as follows:

$$SR = 50\% \times \frac{Score1 - \min(Score1)}{\max(Score1) - \min(Score1)} + 50\% \times \frac{Score2 - \min(Score2)}{\max(Score2) - \min(Score2)} \quad (7)$$

The results of solution accuracy and time usage of each algorithm on the CEC2017 test suites are listed in Table 3 and Table 4. To verify how each algorithm performs in different environments, we calculated the values of *SR* based on the results in Table 3 and Table 4.

Fig. 4 illustrates the comparison of the overall score of each algorithm, *SR*, derived from the solution accuracy and computational efficiency. The results reveal that NGA attains the highest score (92.11 %) among the algorithms, showcasing its superior performance. Following NGA, STA and RCGA secure the second and third positions with scores of 79.19 % and 75.01 %, respectively. However, SA and ACOR demonstrate relatively poorer performance compared to the other algorithms. When considering the scores based on solution accuracy, MAPSO achieves the highest score (50 %), followed by IA41 with the second highest score (47.82 %), and NGA with the third (45.95 %). As for scores based on time usage, STA secures the highest score (50 %), NGA follows with the second highest score (46.15 %), and SA and RCGA claim the third position (35.26 %). Overall, these findings highlight the commendable performance of NGA in both solution accuracy and computational efficiency.

Table 5
Parameter settings of the NGA parameters.

No.	n	P_c	P_m	p	q
Group1	50	0.4	0.15	1	0
Group2	50	0.4	0.15	3	0
Group3	50	0.4	0.15	5	0
Group4	50	0.4	0.15	8	0
Group5	50	0.4	0.15	10	0
Group6	50	0.4	0.15	1	3
Group7	50	0.4	0.15	1	5
Group8	50	0.4	0.15	1	8
Group9	50	0.4	0.15	1	10
Group10	50	0.4	0.15	3	3
Group11	50	0.4	0.15	3	5
Group12	50	0.4	0.15	3	8
Group13	50	0.4	0.15	3	10

Note: p or $q = 3$ is obtained by rounding up the result of 5 %*n; p or $q = 5$ is obtained by calculating 10 %*n; p or $q = 8$ is obtained by rounding up the result of 15 %*n; p or $q = 10$ is obtained by calculating 20 %*n.

3.3. Sensitivity analysis on NGA parameters

The rock discontinuity clustering problem is essentially a two-dimensional optimization problem. In this section, based on the dimension case ($D = 2$), we analyze the sensitivity of the NGA parameters, particularly focusing on the parameters p and q, which represent

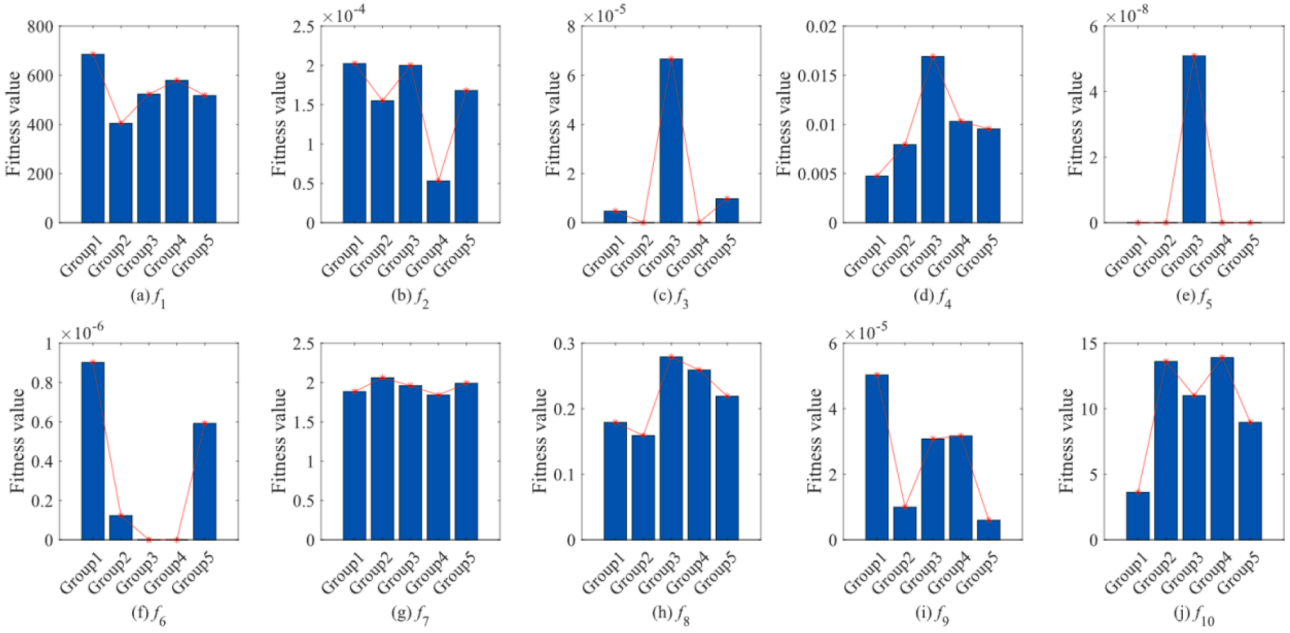


Fig. 5. Comparison of mean values of fitness by changing p.

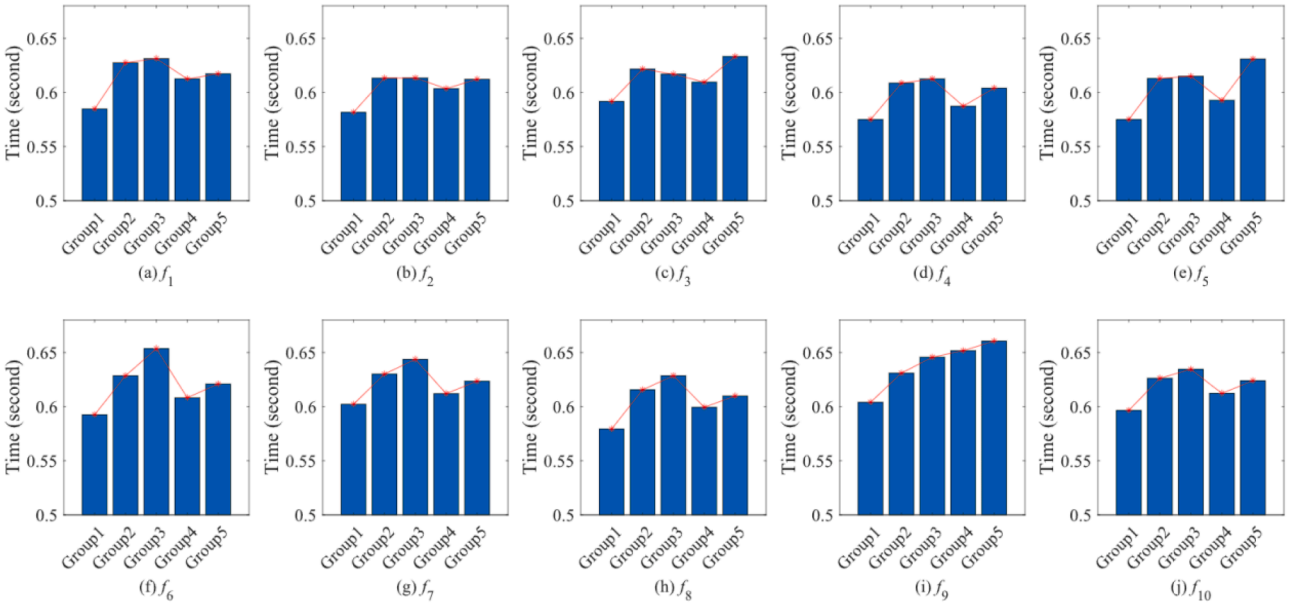


Fig. 6. Comparison of time usage by changing p.

the size of TSS and the size of FSS, respectively. This assessment aims to understand how NGA responds to variations in these key parameters. As shown in Table 5, there are 13 groups of NGA parameter settings. In NGA, if only the optimal solution is considered in the TSS ($p = 1$) and the strategy in the FSS is not considered ($q = 0$), theoretically, NGA at this time (Group1) is equivalent to RCGA. In other words, RCGA can be regarded as a special case of NGA (Group1) according to the definition of NGA introduced in Section 2.2.

The fitness values among the different groups in Fig. 5 appear to be relatively close. However, there doesn't seem to be a general trend by varying p. In other words, as the size of TSS varies, the results obtained from RCGA (Group1) and NGA (Group2 to Group5) remain relatively similar. Additionally, in most cases, the algorithm performs poorly when $p = 10$ %, particularly for f_2, f_3, f_4, f_5, and f_8. Furthermore, as depicted in Fig. 6, the computational time for NGA increases by approximately 2 % to 12 % compared to RCGA (Group1). Nevertheless, the overall increase in computational time for this group is not substantial. It is attributed to the similarity in the complexity of the objective function evaluations between NGA and RCGA, indicating that there is no significant extra computation cost for NGA.

Based on Fig. 7, it is evident that appropriately increasing the value of q (Group6 to Group9) can lead to a reduction in the fitness value. In other words, enlarging the size of FSS can substantially improve the algorithmic performance, which is particularly noticeable in f_4, f_7, f_8, and f_{10}. Additionally, as depicted in Fig. 8, adjusting the value of q allows us to modify the required computation time of NGA (Group6 to Group9), potentially leading to higher computational efficiency compared to RCGA (Group1).

According to Fig. 5 and Fig. 6, we conducted a sensitivity analysis of the NGA parameter p while maintaining $q = 0$. Likewise, based on Fig. 7

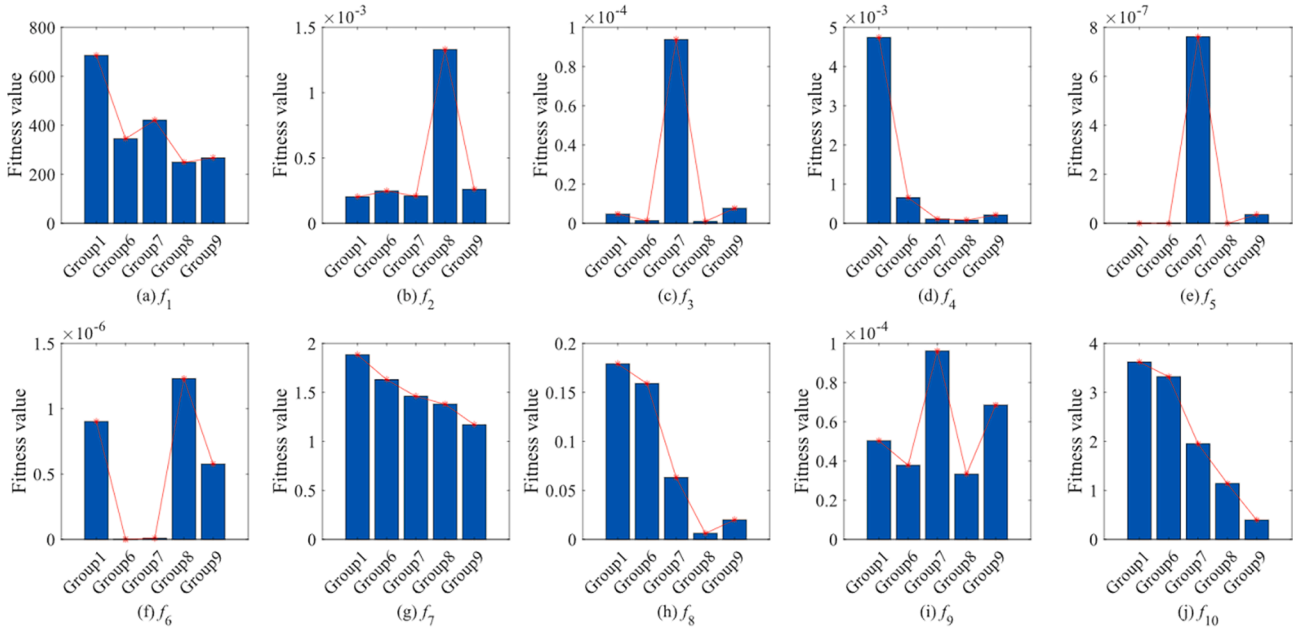


Fig. 7. Comparison of mean values of fitness by changing q.

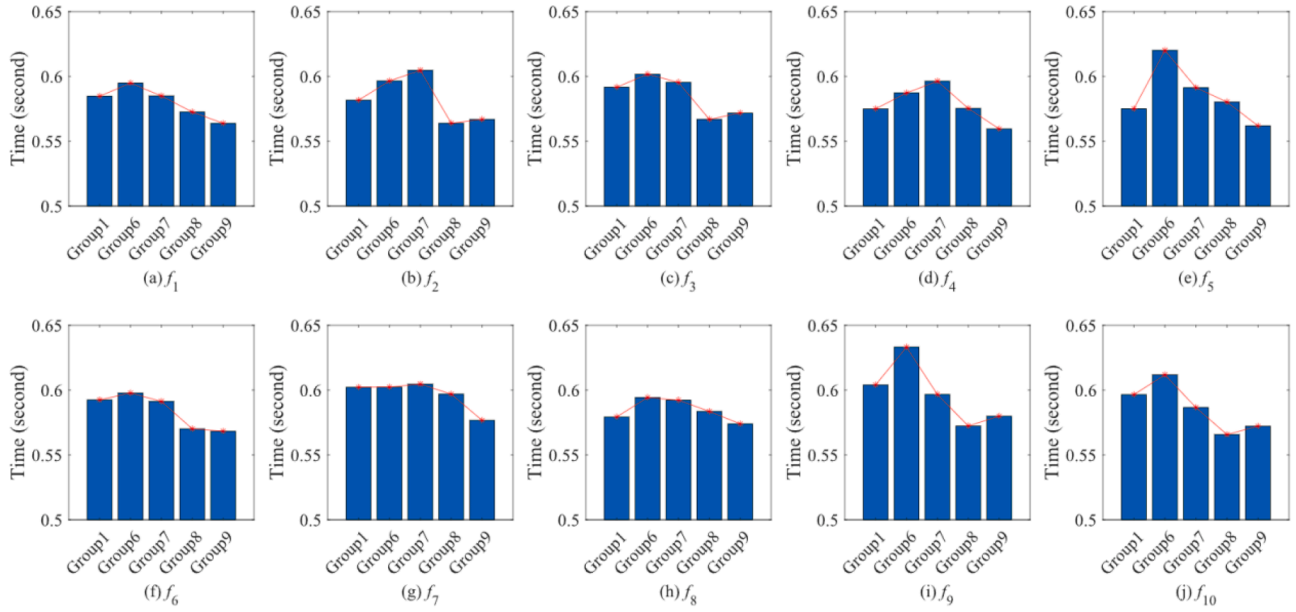


Fig. 8. Comparison of time usage by changing q.

and Fig. 8, we performed a sensitivity analysis of the NGA parameter q while keeping $p = 1$. This isolated analysis based on each NGA parameter has given us a preliminary understanding of the contributions of the TSS and the FSS in NGA.

Next, an experiment comparing Group2, Group10, Group11, and Group12 is conducted to investigate the impact of varying the parameters p and q on the performance of NGA. For comparison, we set $p = 3$, resulting in a TSS size of 3, which enables the access of both the optimal solution and two suboptimal solutions.

Fig. 9 illustrates the impact of increasing q on solution accuracy. Specifically, Group12 exhibits a smaller fitness value compared to the other groups, suggesting that this parameter configuration leads to improved solution accuracy for NGA. However, in contrast to Fig. 7, the fitness value does not exhibit a clear trend as q increases. Meanwhile, Fig. 10 demonstrates that the computational time usage changes with q, echoing the observations from Fig. 8. Simultaneously considering the values of p and q can significantly influence both solution accuracy and computational efficiency. By carefully selecting specific parameters, it can enhance the performance of NGA.

Furthermore, Group12 was taken as an example to analyze the operation convergence of the optimal solution $t_{1,j}$, and suboptimal solutions $t_{2,j}$ and $t_{3,j}$ in the TSS. As shown in Fig. 11, the convergence efficiency of the optimal solution $t_{1,j}$ is better than that of the suboptimal solution $t_{2,j}$, and the convergence efficiency of the suboptimal solution $t_{2,j}$ is better than $t_{3,j}$. It is notable that after the iteration number exceeds 23, both the optimal and suboptimal solutions have converged, with their values being very close to each other. Again, the effectiveness of NGA is verified.

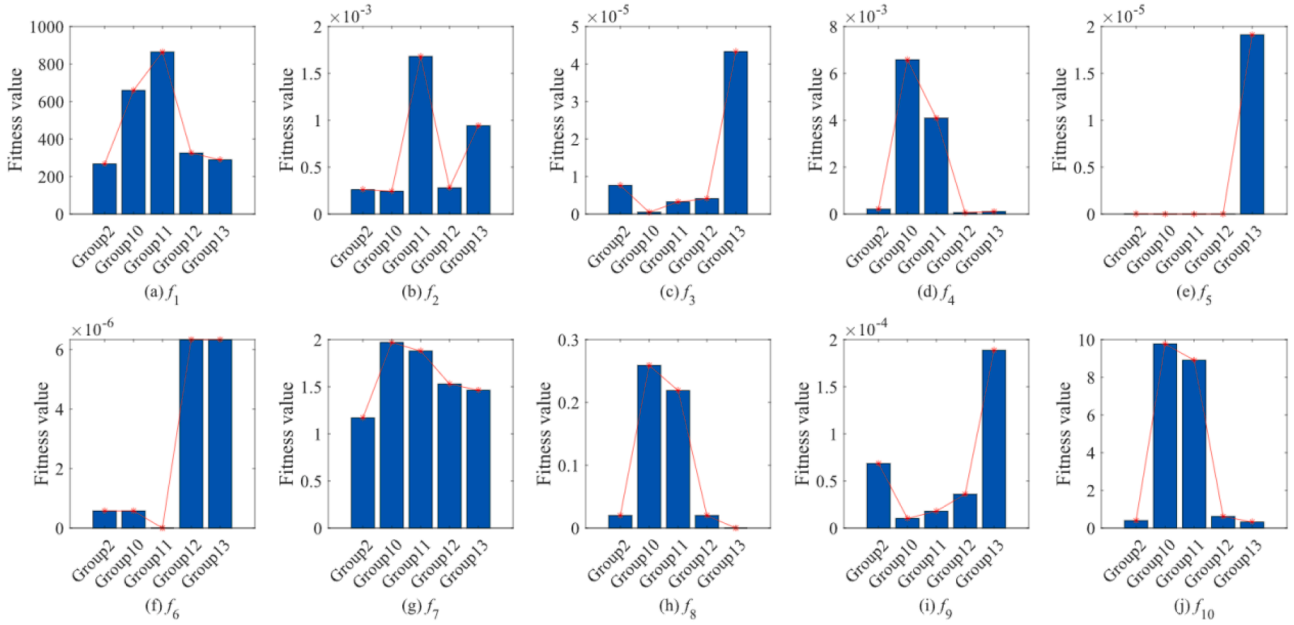


Fig. 9. Comparison of mean values of fitness by changing q in the context of optimal and suboptimal solutions.

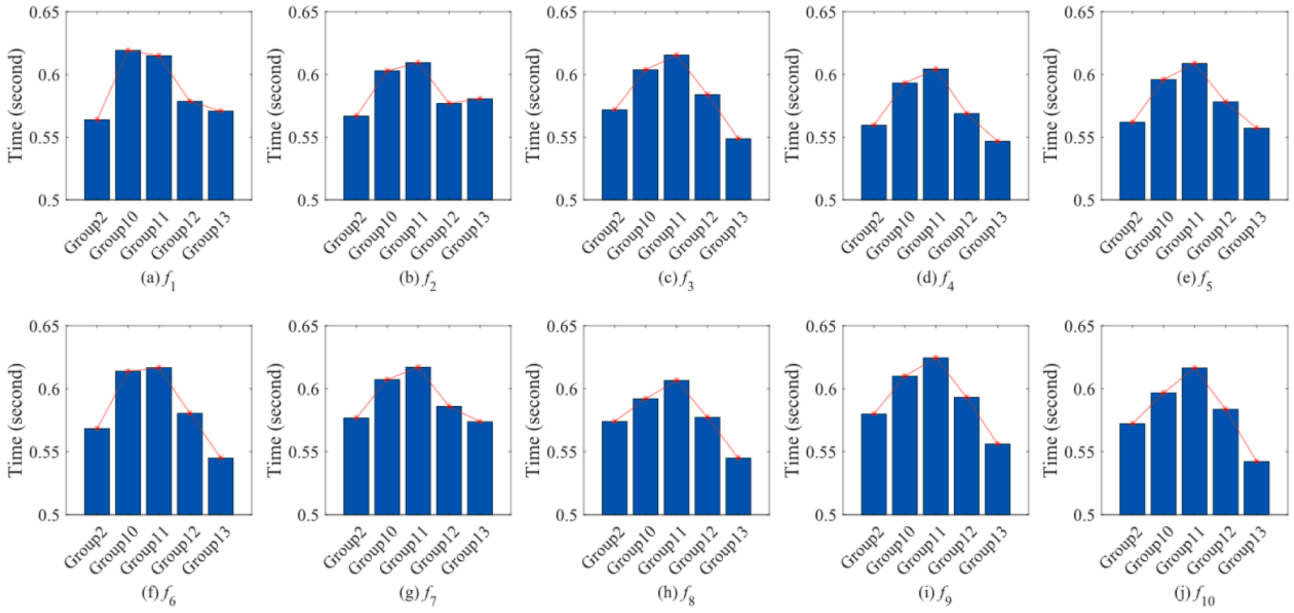


Fig. 10. Comparison of time usage changing q in the context of optimal and suboptimal solutions.

4. NGA for clustering of rock discontinuity sets

4.1. Background

The orientation of a rock discontinuity is usually represented by the dip direction α_d and the dip angle β_d. For the mathematical analysis of the discontinuity orientations, it is usually assumed that the discontinuity surface is a spatial plane. In the right-handed Cartesian coordinate system, as shown in Fig. 12, the positive x-axis is horizontal to the north; the positive y-axis is horizontal to the east and the positive z-axis is vertically downward. The trend α of the normal vector is the angle between the positive x-axis and the projection of the dip direction vector on the x-y plane, measured in clockwise rotation from the north. The plunge β is the angle between the pole vector and the x-y plane. Then, α and β are related to α_d and β_d as follows:$\alpha = \alpha_d \pm 180^\circ (0^\circ \leqslant \alpha \leqslant 360^\circ)$ and $\beta = 90^\circ - \beta_d (0^\circ \leqslant \beta \leqslant 90^\circ)$. The orientation is expressed in terms of its unit normal vector $\boldsymbol{\Theta} = (\alpha, \beta)^T$.

The orientation can be represented by unit normal vector $\boldsymbol{X} = (x, y, z)^{\mathrm{T}}$, where

$$\begin{cases} x = \cos\alpha\cos\beta, \\ y = \sin\alpha\cos\beta, \\ z = \sin\beta, \\ x^2 + y^2 + z^2 = 1. \end{cases} \tag{8}$$

The lower hemisphere projection is used to represent the orientation of the discontinuities. In this projection method, the unit vector of the rock discontinuity is downward, so the endpoint A is located in the lower hemisphere. A straight line is drawn from the upper pole P to point A, which intersects the x-y plane (stereographic plane) at point A'. The projection A' in the x-y plane is called a polar stereographic, or

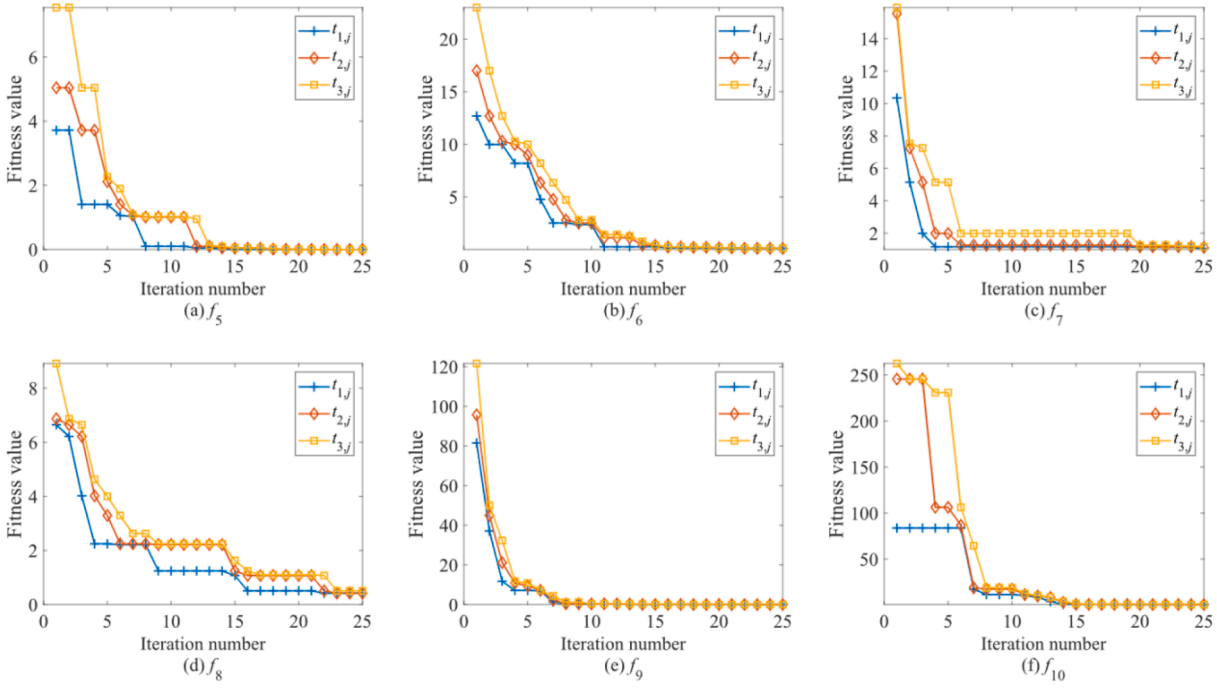


Fig. 11. Convergence results of the optimal and suboptimal solutions.

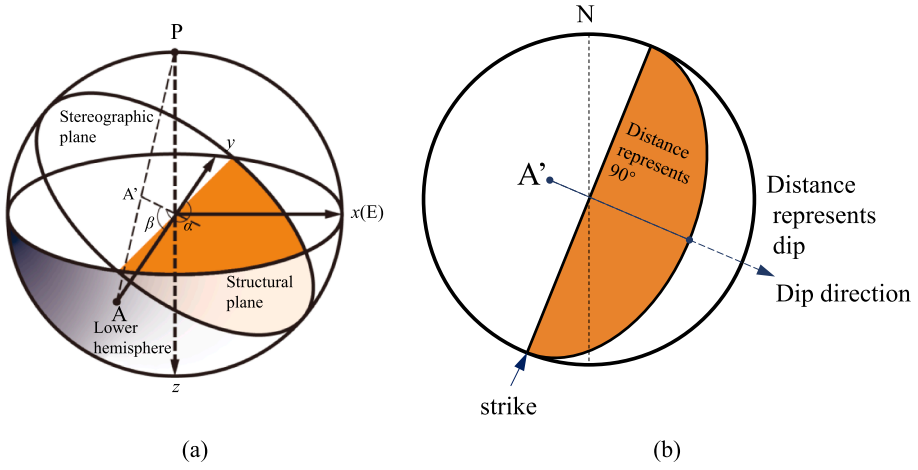


Fig. 12. Spatial presentation of discontinuity orientations (N stands for North; E stands for East) (a) spherical projection; (b) plane view of hemispherical projection (Liu et al., 2017).

hemispherical plot Fig. 12.

In the case of the K-means clustering method, the commonly used method is Euclidean distance to measure the distance between the clustering center and the data points within the cluster. However, this approach may lead to the separation of closely located discontinuities into different sets. For instance, consider two closely spaced discontinuities with a steep angle of inclination close to 90°. If their orientations differ by approximately 180°, they should be grouped together, but using Euclidean distance alone may result in their division into separate clusters.

The utilized sine-squared measure could avoid the situation and be more efficient. The angle between two discontinuities is equal to the angle between their corresponding unit normal vectors $X_1 = (x_1, y_1, z_1)$ and $X_2 = (x_2, y_2, z_2)$, which can be expressed as

$$\theta = \arccos\left|X_1 \cdot X_2^T\right| \tag{9}$$

Based on Eq. (9), the sine-squared similarity measure becomes

$$D = \sin^2\theta = 1 - \left(X_1 \cdot X_2^T\right)^2 \tag{10}$$

Suppose that N discontinuities P_j ($j = 1, 2, 3, \ldots, N$) are divided into C groups, and the clustering center of each group is V_i ($i = 1, 2, 3, \ldots, C$). The distance $D(P_j, V_j)$ between P_j and V_i is obtained by Eq. (11), defining $d_{i,j}$ as the degree of affiliation of the j-th discontinuity belonging to the i-th clustering center:

$$d_{ij} = \frac{1}{D(P_j, V_j)}\left[\sum_{k=1}^{C}\left(\frac{1}{D(P_j, V_k)}\right)\right]^{-1} \tag{11}$$

The distance between all clustering centers and each discontinuity can be summarized as follows:

$$\text{MIN } J = \sum_{j=1}^{N}\sum_{i=1}^{C} d_{ij}^2 D\left(P_j, V_i\right) \tag{12}$$

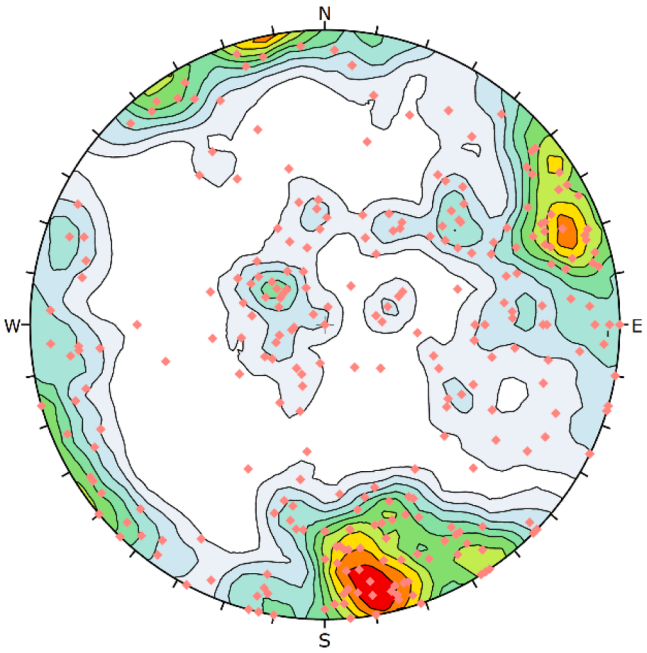


Fig. 13. Pole and contour plots of discontinuities in the data set by Shanley & Mahtab (1976).

where J is the target function in this optimization.

To assess the effectiveness of the clustering, we employed the Calinski-Harabasz Index (CHI) (Calinski & Harabasz, 1974) as described in Eq. (13). It is a widely adopted evaluation metric for clustering analysis, specifically designed to evaluate the quality of clustering outcomes. It takes into account both the dispersion within clusters and the separation between clusters.

$$CHI = \frac{B}{W} \times \frac{N-k}{k-1} \tag{13}$$

where B represents the between-cluster dispersion, which is the sum of squared distances between cluster centroids and the overall centroid, W represents the within-cluster dispersion, which is the sum of squared distances between data points and their respective cluster centroids, N is the total number of data points, and k is the number of clusters.

Eq. (13) measures the ratio of between-cluster dispersion to within-cluster dispersion, taking into account both the number of data points and the number of clusters. A higher value of CHI indicates a better clustering result, indicating improved separation between clusters and compactness within each cluster.

4.2. NGA optimized K-means clustering method

Based on the proposed NGA in this study, a new method is suggested for clustering rock discontinuity sets. The procedure of this method is provided as the following steps:

Step 1: Preprocess the input data and generate the initial population using Eq. (1).

Step 2: Iterate in the population and calculate the fitness of each individual based on Eq. (12).

Step 3: Start NGA and apply its operators by using Eqs. (3) – (6).

Step 4: Apply the optimal cluster centers found in NGA to the K-means algorithm to obtain the final clustering results.

To verify this method, the data set by Shanley & Mahtab (1976) is taken as an example. It contains 286 discontinuities sampled at the San Manuel copper mine in Arizona, USA. Fig. 13 shows the pole and contour plots of discontinuities.

This dataset has been widely utilized in studies focusing on clustering discontinuity sets. The pioneering work by Shanley & Mahtab (1976), referred to as the S&M method in this study, introduced the first published approach for discontinuity clustering, which remains highly influential in the field. In our study, we compare our results with those obtained from alternative approaches, including Fuzzy Clustering Method (FCM) (Hammah & Curran, 1998), Spectral Clustering (SC) (Jimenez-Rodriguez & Sitar, 2006), K-means method based on particle swarm optimization (KPSO) (Li et al., 2015), and the Affinity Propagation (AP) algorithm (Liu et al., 2017).

Table 6
Clustering results of different methods based on the data set by Shanley & Mahtab (1976).

Method	set	Number of discontinuities	CHI	Time/s
NGA-based K-means clustering method	1 (●)	121	3.1358	1.219
	2 (▲)	115		
	3 (◆)	50		
AP (Liu et al., 2017)	1 (●)	114	2.1144	2.159
	2 (▲)	111		
	3 (◆)	61		
SC (Jimenez-Rodriguez & Sitar, 2006)	1 (●)	152	1.6398	0.715
	2 (▲)	79		
	3 (◆)	59		
FCM (Hammah & Curran, 1998)	1 (●)	121	2.1465	1.077
	2 (▲)	93		
	3 (◆)	72		
KPSO (Li et al., 2015)	1 (●)	110	1.9958	13.201
	2 (▲)	110		
	3 (◆)	66		
S&M method (Shanley & Mahtab, 1976)	1 (●)	114	1.6555	1.169
	2 (▲)	114		
	3 (◆)	58		

We conducted a comparison of the CPU time required by different algorithms, ensuring that all methods were tested on the same platform and software setup. The experiments were carried out on a Windows 10 system running MATLAB 2021b, equipped with an Intel(R) Core (TM) i5-12400f @3.3 GHz CPU and 16.0 GB of RAM. Table 6 provides an overview of the assignment situation and computation times for each method. The clustering results are graphically presented in pole charts, as illustrated in Fig. 14.

Evaluating the clustering results obtained from different algorithms (Fig. 14), it is worth noting the distinct patterns exhibited by the SC algorithm compared to other methods. Despite having the lowest CPU time, its clustering results are comparatively poor, and it gets the lowest CHI of 1.6398. On the other hand, the remaining algorithms demonstrate similar grouping boundaries. Among these algorithms, NGA demonstrates the highest fidelity to the original grouping, achieving a CHI score of 3.1358 and showcasing fast computation speed. The widely-used AP algorithm has a balance between accuracy and time efficiency, ranking second highest in terms of its 2.1144 CHI score among the algorithms. FCM, which requires determining initial clustering centers, yields a CHI score of 2.1465, similar to AP, and

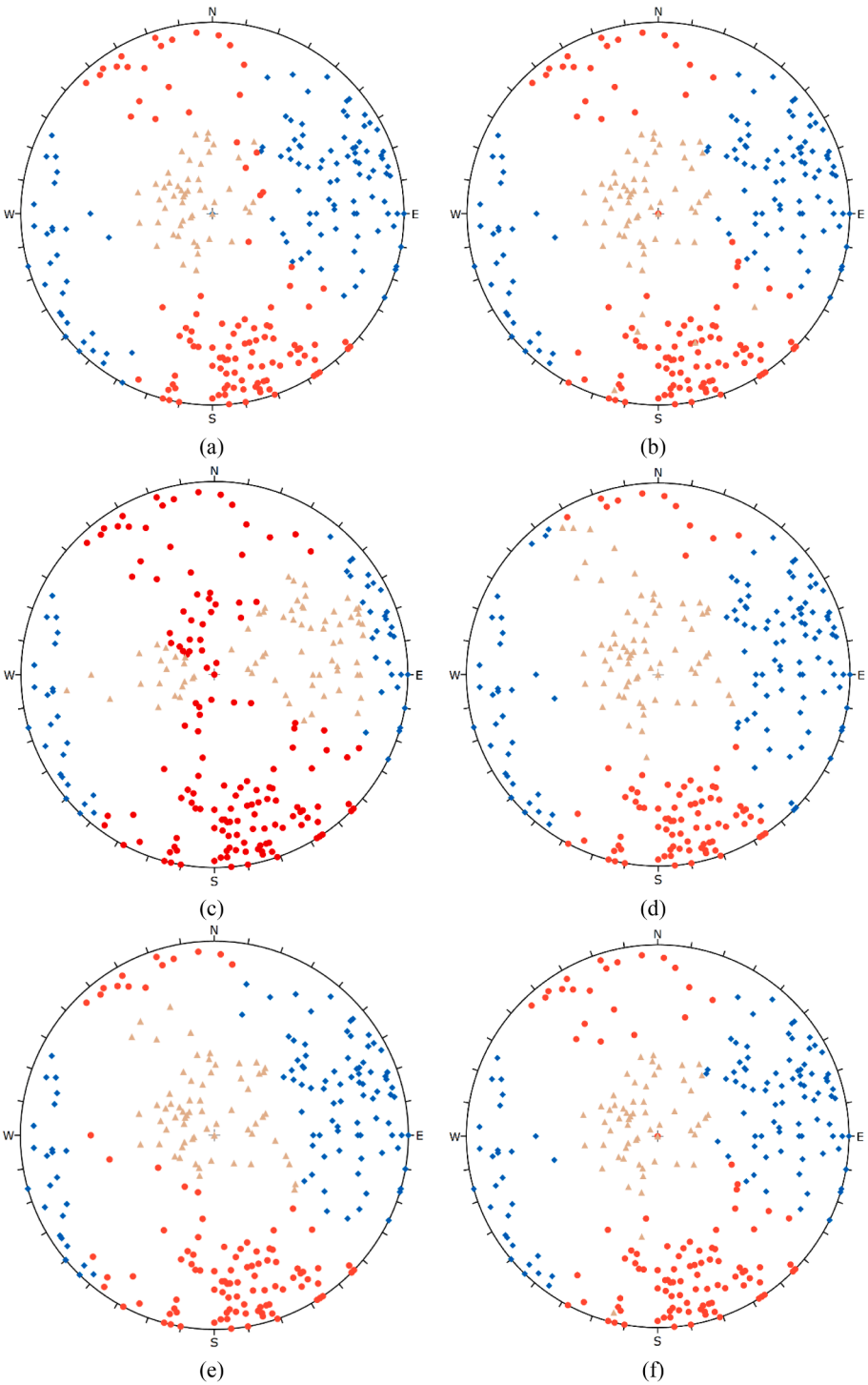


Fig. 14. Comparison of the clustering results for the Shanley and Mahtab data: (a) NGA-based K-means clustering method; (b) AP; (c) SC; (d) FCM; (e) KPSO; (f) S&M method.

demonstrates low computation time. The KPSO algorithm requires the most CPU time and produces 1.9958 CHI scores compared to NGA, FCM, and AP. The classic S&M method gets a CHI of 1.6555, it seems not very suitable for this dataset. Thus, according to this example, NGA not only demonstrates high calculation accuracy but also competes with other algorithms in terms of time efficiency.

5. Case study

The Lanping open-pit mine is located in Yunnan Province, China, with geographical coordinates of 99°25′39.47″E and 26°24′13.30″N (Fig. 15). The highest elevation of the mine area is 2885 m, and the lowest elevation is 2475 m, with a relative height difference of 410 m, which is a low-mountain terrain. After years of mining operations, the open-pit mine forms a reverse "C" shape. In this study, the data of 117 discontinuities were observed from various locations in the south of the open-pit mine. The countered pole of these sampled discontinuities is shown in Fig. 16. The NGA-optimized K-means method was used to identify the discontinuity sets using these data.

A comparison of the results is performed based on the proposed

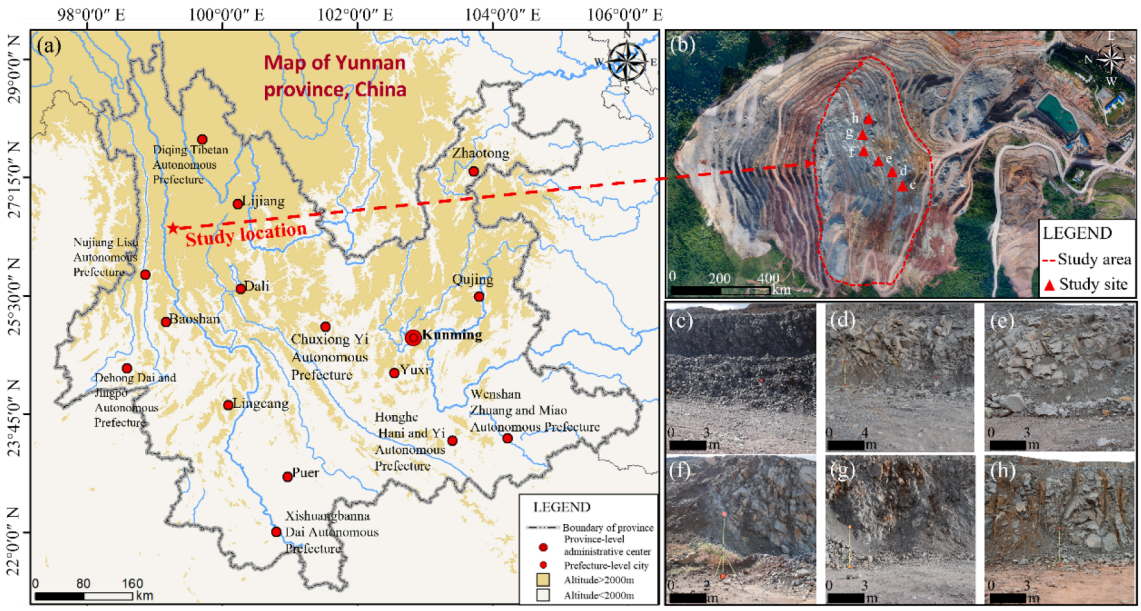


Fig. 15. The open-pit mine slope studied: (a) Location of the open-pit mine slope; (b) Image of the Lanping open-pit mine; (c) - (f) Images of the study sites.

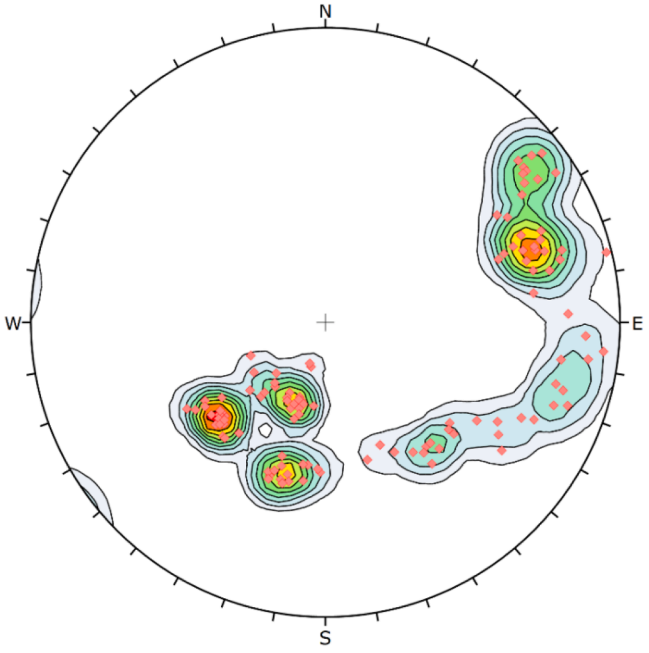


Fig. 16. Countered pole of discontinuities sampled in study sites.

method and other clustering methods. The results are shown in Fig. 17 and Table 7.

As shown in Fig. 17, NGA exhibits three distinct and well-separated clusters, achieving the highest CHI (7.0620) among all algorithms. AP, FCM, and KPSO generate cluster boundaries that are identical to each other, with corresponding CHI scores of 2.7465, 3.3920, and 2.7409, respectively. SC remains the fastest algorithm in terms of computation time, yet it yields the lowest CHI score of 1.8654. Interestingly, the SH method proves to be suitable for our dataset, offering a CHI score of 5.1992, ranking third among the algorithms.

6. Conclusion

This study presented a novel NGA that integrates the neutrosophic concept into genetic algorithms. NGA introduced three distinct solution spaces (TSS, FSS, ISS) based on the notions of true, false, and indeterminate information. Unlike RCGAs, NGA offers a flexible population initialization framework, while its subsequent evolutionary processes catered to different individual characteristics based on fitness values within each solution space. NGA introduced fine-tuning in the TSS, adaptive regeneration in the FSS, and modified crossover and mutation operations in the ISS, enhancing its ability to navigate away from local optima while reducing computational complexity.

NGA was evaluated against several prior algorithms, such as MAPSO, IA, STA, PSO, SA, ACOR, and RCGA, using the CEC2017 test suites to assess its performance across various problems and conditions. The comparison was based on solution accuracy and computational efficiency, with NGA showcasing superior performance, attaining the highest overall score (92.11 %) among the algorithms. Additionally, sensitivity analysis of the NGA parameters p and q, particularly focusing on the size of TSS and FSS, revealed valuable insights into algorithm performance variations. Varying the parameter p did not reveal a general trend, indicating that the results obtained from RCGA and NGA remained relatively similar across different TSS sizes. Notably, the computational time for NGA increased by approximately 2 % to 12 % compared to RCGA. Moreover, appropriately increasing the value of q led to a reduction in fitness value, suggesting that enlarging the size of FSS substantially improved the algorithm's performance. Furthermore, an experiment investigating the impact of varying parameters p and q revealed that considering both parameters simultaneously can significantly influence solution accuracy and computational efficiency. This comprehensive analysis underscored the critical role of parameter selection in enhancing the performance of NGA while providing valuable insights into the convergence properties of optimal and suboptimal solutions within the TSS.

In the application for the rock discontinuity clustering problem, the NGA-optimized K-means method was used. Using the data set of Shanley & Mahtab (1976) for testing, we compared it with the AP, SC, FCM, KPSO and S&M methods. We used CHI for evaluation. The method proposed in this paper had a very good CHI index of 3.1358, which was much better than the other algorithms. The CPU time for this method was 1.219 s. The result was better than the same type of iterative algorithm by over 57.11 %. Additionally, a data set collected from the Lanping open-pit mine was used for testing and comparing the proposed NGA with the AP, SC, FCM, KPSO, and S&M methods. The CHI of the

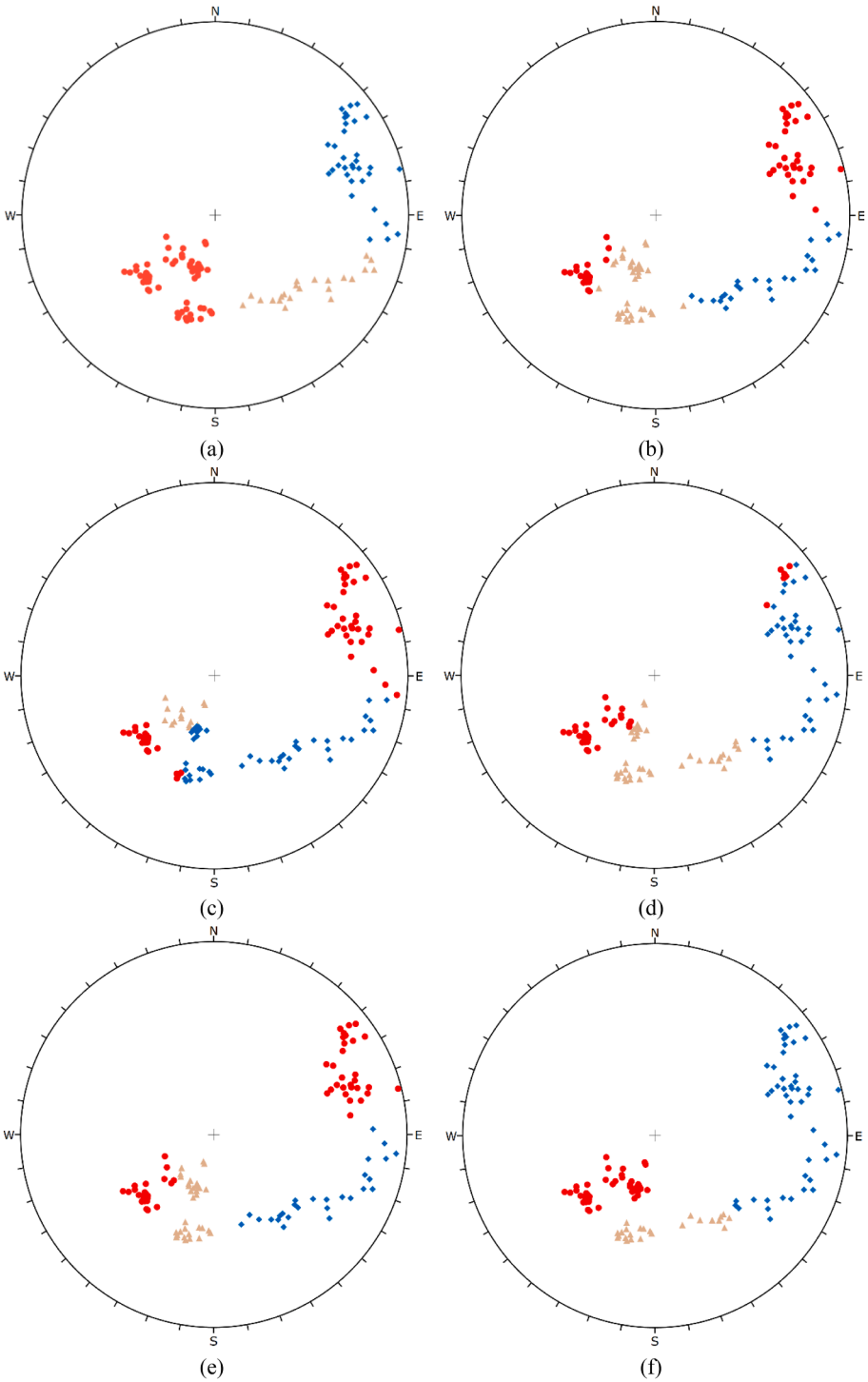


Fig. 17. Comparison of the clustering results for the data sampled from study sites. (a) NGA-based K-means clustering method; (b) AP; (c) SC; (d) FCM; (e) KPSO; (f) S&M method.

proposed method was 7.0620, which was the highest compared with other algorithms, and its computational efficiency was maintained, the CPU time of this method was 0.681 s, which was faster than the same type of iterative algorithms by more than 10.16 %.

This study's limitations should also be addressed. Firstly, while the sensitivity analysis on the NGA parameters highlights the significant impact of the TSS size and FSS size on NGA performance, further investigation is needed to determine the optimal settings for p and q across different problem domains. Secondly, the operations in the ISS rely on classical crossover and mutation operators. Introducing advanced strategies, such as new selection criterion functions to trigger crossover and mutation, could potentially enhance NGA performance. This enhancement would allow the operations of the ISS to make a more substantial contribution to NGA. Thirdly, our application of NGA to rock discontinuity set clustering analysis does not fully showcase its strengths in handling problems that require consideration of both optimal and suboptimal solutions. Future studies will focus on exploring more complex application scenarios for NGA, such as multi-peak optimization problems, to better demonstrate its capabilities.

Table 7
Clustering results of different methods based on the Lanping open-pit mine dataset.

Method	set	Number of discontinuities	CHI	Time/s
NGA-based K-means clustering method	1 (●)	58	7.0620	0.681
	2 (▲)	23		
	3 (◆)	36		
AP (Liu et al., 2017)	1 (●)	83	2.7465	0.618
	2 (▲)	26		
	3 (◆)	38		
SC (Jimenez-Rodriguez & Sitar, 2006)	1 (●)	58	1.8654	0.089
	2 (▲)	45		
	3 (◆)	14		
FCM (Hammah & Curran, 1998)	1 (●)	36	3.3920	0.776
	2 (▲)	41		
	3 (◆)	40		
KPSO (Li et al., 2015)	1 (●)	55	2.7409	13.535
	2 (▲)	34		
	3 (◆)	28		
S&M method (Shanley & Mahtab, 1976)	1 (●)	50	5.1992	0.499
	2 (▲)	42		
	3 (◆)	25		

CRediT authorship contribution statement

Rui Yong: Data curation, Methodology, Software, Writing – original draft. **Hanzhong Wang:** Investigation, Resources, Formal analysis. **Jun Ye:** Conceptualization, Supervision. **Shigui Du:** Validation, Writing – review & editing. **Zhanyou Luo:** Investigation, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The study was funded by the National Natural Science Foundation of China (No. 42177117 and No. 42277147).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2023.122973. The code for the Neutrosophic Genetic Algorithm associated with the reproducible capsule can be accessed via the following link: https://doi.org/10.24433/CO.1557783.v1. The code for the NGA-K means associated with the reproducible capsule can be accessed via the following link: https://doi.org/10.24433/CO.9606342.v1.

References

Almabsout, E. A., El-Sehiemy, R. A., & Bayoumi, A. S. A. (2023). Enhanced real coded genetic algorithm for optimal DG placement in a radial distribution system. *Journal of Electrical Engineering & Technology, 18*(4), 2581–2597. https://doi.org/10.1007/s42835-022-01355-1

Ashour, A. S., Hawas, A. R., Guo, Y., & Wahba, M. A. (2018). A novel optimized neutrosophic k-means using genetic algorithm for skin lesion detection in dermoscopy images. *Signal, Image and Video Processing, 12*(7), 1311–1318.

Awad, N. H., Ali, M. Z., Suganthan, P. N., Liang, J. J., & Qu, B. Y. (2017). *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*.

Bonyadi, M. R., & Michalewicz, Z. (2017). Particle swarm optimization for single objective continuous space problems: A review. *Evolutionary Computation, 25*(1), 1–54. https://doi.org/10.1162/EVCO_r_00180

Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods, 3*(1), 1–27. https://doi.org/10.1080/03610927408827101

Carrasco, J., García, S., Rueda, M., Das, S., & Herrera, F. (2020). Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm and Evolutionary Computation, 54*, Article 100665.

Cui, X., & Yan, E. (2020). Fuzzy C-means cluster analysis based on variable length string genetic algorithm for the grouping of rock discontinuity sets. *KSCE Journal of Civil Engineering, 24*, 3237–3246.

Du, S.-G., Saroglou, C., Chen, Y., Lin, H., & Yong, R. (2022). A new approach for evaluation of slope stability in large open-pit mines: A case study at the Dexing Copper Mine, China. *Environmental Earth Sciences, 81*(3), 102.

Elwahsh, H., Gamal, M., Salama, A. A., & El-Henawy, I. M. (2018). A novel approach for classifying Manets attacks with a neutrosophic intelligent system based on genetic algorithm. *Security and Communication Networks, 2018*.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena, 22*(1), 187–204. https://doi.org/10.1016/0167-2789(86)90240-X

Hammah, R., & Curran, J. (1998). Fuzzy cluster algorithm for the automatic identification of joint sets. *International Journal of Rock Mechanics and Mining Sciences, 35*(7), 889–905.

Holland, J. H. (1992). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. *The MIT Press*. https://doi.org/10.7551/mitpress/1090.001.0001

Huang, L., Tang, H., Wang, L., & Juang, C. (2019). Minimum scanline-to-fracture angle and sample size required to produce a highly accurate estimate of the 3-D fracture orientation distribution. *Rock Mechanics and Rock Engineering, 52*, 803–825.

Jimenez, R. (2008). Fuzzy spectral clustering for identification of rock discontinuity sets. *Rock Mechanics and Rock Engineering, 41*(6), 929.

Jimenez-Rodriguez, R., & Sitar, N. (2006). A spectral method for clustering of rock discontinuity sets. *International Journal of Rock Mechanics and Mining Sciences, 43*(7), 1052–1061.

Kemeny, J., & Post, R. (2003). Estimating three-dimensional rock discontinuity orientation from digital images of fracture traces. *Computers & Geosciences, 29*(1), 65–77.

Kirkpatrick, S., Gelatt, C. D., Jr, & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.

Kumar, M., Bhutani, K., & Aggarwal, S. (2015). Hybrid model for medical diagnosis using Neutrosophic Cognitive Maps with Genetic Algorithms. *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–7.

Li, D., Wang, J., Sun, W., & Zhang, N. (2023). Matrix non-structural model and its application in heat exchanger network without stream split. *Processes, 11*(6), 1843. https://doi.org/10.3390/pr11061843

Li, Y., Wang, Q., Chen, J., Xu, L., & Song, S. (2015). K-means algorithm based on particle swarm optimization for the identification of rock discontinuity sets. *Rock Mechanics and Rock Engineering, 48*(1), 375–385. https://doi.org/10.1007/s00603-014-0569-x

Liu, J., Zhao, X.-D., & Xu, Z. (2017). Identification of rock discontinuity sets based on a modified affinity propagation algorithm. *International Journal of Rock Mechanics and Mining Sciences, 94*, 32–42. https://doi.org/10.1016/j.ijrmms.2017.02.012

Lu, B., Ding, X., & Wu, A. (2007). Study on method of orientation data partitioning of randomly distributed discontinuities of rocks. *Yanshilixue Yu Gongcheng Xuebao/Chinese Journal of Rock Mechanics and Engineering, 26*(9), 1809–1816.

Michalewicz, Z., Janikow, C. Z., & Krawczyk, J. B. (1992). A modified genetic algorithm for optimal control problems. *Computers & Mathematics with Applications, 23*(12), 83–94. https://doi.org/10.1016/0898-1221(92)90094-X

Moorthy, V., & Marappan, K. (2022). Identification of delamination severity in a tapered FRP composite plate. *Composite Structures, 299*, Article 116054.

Narang, P., & De, P. K. (2023). An imperfect production-inventory model for reworked items with advertisement, time and price dependent demand for non-instantaneous deteriorating item using genetic algorithm. *International Journal of Mathematics in Operational Research, 24*(1), 53–77.

Shanley, R. J., & Mahtab, M. A. (1976). Delineation and analysis of clusters in orientation data. *Journal of the International Association for Mathematical Geology, 8*(1), 9–23. https://doi.org/10.1007/BF01039681

Smarandache, F. (1998). *Neutrosophy: Neutrosophic probability, set, and logic*. Rehoboth, USA: American Research Press.

Song, H., Wang, J., Song, L., Zhang, H., Bei, J., Ni, J., & Ye, B. (2022). Improvement and application of hybrid real-coded genetic algorithm. *Applied Intelligence, 52*(15), 17410–17448. https://doi.org/10.1007/s10489-021-03048-0

Song, S., Wang, Q., Chen, J., Li, Y., Zhang, W., & Ruan, Y. (2017). Fuzzy C-means clustering analysis based on quantum particle swarm optimization algorithm for the grouping of rock discontinuity sets. *KSCE Journal of Civil Engineering, 21*(4), 1115–1122. https://doi.org/10.1007/s12205-016-1223-9

Tokhmechi, B., Memarian, H., Moshiri, B., Rasouli, V., & Noubari, H. A. (2011). Investigating the validity of conventional joint set clustering methods. *Engineering Geology, 118*(3–4), 75–81.

Wei, B., Xia, X., Yu, F., Zhang, Y., Xu, X., Wu, H., Gui, L., & He, G. (2020). Multiple adaptive strategies based particle swarm optimization algorithm. *Swarm and Evolutionary Computation, 57*, Article 100731. https://doi.org/10.1016/j.swevo.2020.100731

Wei, C., & Cheng, J. (2022). Optimal allocation of irrigation water in a single-reservoir and a single-pumping-station system under deficit irrigation conditions. *Water Supply, 22*(12), 8418–8433. https://doi.org/10.2166/ws.2022.424

Wu, Q., & Kulatilake, P. (2012). REV and its properties on fracture system and mechanical properties, and an orthotropic constitutive model for a jointed rock mass in a dam site in China. *Computers and Geotechnics, 43*, 124–142.

Ye, J. (2019). PID tuning method using single-valued neutrosophic cosine measure and genetic algorithm. *Intelligent Automation and Soft Computing, 25*(1), 15–23.

Zhang, L., Shen, H., Xu, K., Huang, W., Wang, Y., Chen, M., & Han, B. (2023). Effect of ceramic waste tile as a fine aggregate on the mechanical properties of low-carbon ultrahigh performance concrete. *Construction and Building Materials, 370*, Article 130595. https://doi.org/10.1016/j.conbuildmat.2023.130595

Zhou, W., & Maerz, N. H. (2002). Implementation of multivariate clustering methods for characterizing discontinuities data from scanlines and oriented boreholes. *Computers & Geosciences, 28*(7), 827–839.

Zhou, X., Gui, W., Heidari, A. A., Cai, Z., Liang, G., & Chen, H. (2023). Random following ant colony optimization: Continuous and binary variants for global optimization and feature selection. *Applied Soft Computing, 144*, Article 110513. https://doi.org/10.1016/j.asoc.2023.110513

Zhou, X., Tian, J., Wang, Z., Yang, C., Huang, T., & Xu, X. (2022). Nonlinear bilevel programming approach for decentralized supply chain using a hybrid state transition algorithm. *Knowledge-Based Systems, 240*, Article 108119. https://doi.org/10.1016/j.knosys.2022.108119