scientific reports

Check for updates

OPEN

Study on the anti-penetration randomness of metal protective structures based on optimized artificial neural network

Lan Liu[1]✉, Weidong Chen[1], Shengzhuo Lu[1]✉, Yanchun Yu[2] & Mingwu Sun[1]

In order to study the Anti-Penetration Randomness of Metal Protective Structures (APRMPS) for the penetration probabilities of Metal Protective Structures under the action of the basic random variables, this paper analyzes the candidates for the basic random variables and the random response of APRMPS, and, on the basis of the improvement of Genetic Algorithm, proposes Dynamic Lifecycle Genetic Algorithm, including its main processes of the optimization of Back Propagation Neural Network. And by adopting the Back Propagation Neural Network optimized by Dynamic Lifecycle Genetic Algorithm (DLGABPNN) as the surrogate model of APRMPS, this paper presents the technical route of DLGABPNN-MCS, the Monte Carlo Simulation with DLGABPNN calculation as repeated sampling tests, to addressing APRMPS. Finally, with two applied examples of the anti-penetration randomness of metal targets, this paper demonstrates the application procedures for this method, proves the higher efficiency of Dynamic Lifecycle Genetic Algorithm than Genetic Algorithm in optimizing Back Propagation Neural Network and verifies the effectiveness of DLGABPNN-MCS in studying APRMPS. This paper may have some significance in proposing Dynamic Lifecycle Genetic Algorithm—the new, universal heuristic algorithm, and providing the common technical method for the APRMPS study based on DLGABPNN-MCS, in the hope of promoting the application of Dynamic Lifecycle Genetic Algorithm in other optimization problems, and offering reference for the further study of APRMPS or the study of other random problems.

Keywords Metal protective structure, Anti-penetration randomness, Penetration probability, Dynamic lifecycle genetic algorithm, Back propagation neural network (BPNN), Surrogate model, Monte Carlo simulation (MCS)

The anti-penetration of protective structures is a typical transient high-strain problem. Since any slight change in conditions will lead to those in the penetration results of protective structures, in the anti-penetration tests of protective structures, random results will occur as for whether the protective structures at the penetration limit state can be penetrated by the fragments. This explains why some researches labeled the ballistic limit velocity as V_{50}[1–4], indicating that the fragments, at the ballistic limit velocity, will penetrate the protective structures with a 50% probability. Although it has been repeatedly found that there is uncertainty in the mechanical behavior of the anti-penetration of protective structures, the anti-penetration of protective structures has been taken as a deterministic one, leading to the ignorance about the mechanical behavior uncertainty of the anti-penetration of protective structures. Under this circumstance, it is necessary to perceive the anti-penetration of protective structures as a randomness problem, and to make a stochastic analysis of the randomness of the mechanical behavior of the anti-penetration of protective structures, so as to obtain the penetration probabilities of protective structures under the action of basic random variables.

What should also be noted is that machine learning has found extensive use in engineering. Khatir et al.[5] employed Firefly Algorith and Genetic Algorithm (GA) to optimize the coordinate modal assurance criterion for complex structures, so as to identify multiple local damages in complex structures. Khatir et al.[6] used Particle Swarm Optimization Algorithm (PSO) and GA in optimizing the gradient boosting for strain prediction in near-surface mounted of fiber-reinforced polymer strengthened reinforced concrete beam, thus achieving the

[1]College of Aerospace and Civil Engineering, Harbin Engineering University, Harbin 150001, China. [2]School of Water Conservancy and Civil Engineering, Northeast Agricultural University, Harbin 150030, China. ✉email: wangyi_liulan@163.com; lusz@hrbeu.edu.cn

accurate result of strain prediction. Khatir et al.[7] even applied the Artificial Neural Network (ANN) optimized by Grasshopper Optimization Algorithm in predicting multiple damages represented by holes in the aluminum plate, and found that the ANNs optimized in this way demonstrated superior performance in damage forecasting. Also, Khatir et al.[8] constructed the hybrid ANN-RSA model by optimizing ANN with Reptile Search Algorithm (RSA) for structural damage detection in steel beams, finding that "the integration of RSA into ANN training significantly improves the accuracy and computational efficiency of damage prediction in structural health monitoring." Besides, Khatir et al.[9] proposed a new hybrid algorithm by integrating PSO into YUKI, and further combined the PSO-YUKI algorithm with radial basis functions for double cracks identification in carbon fiber reinforced polymer cantilever beams. Their results show that the PSO-YUKI algorithm shows robustness in double cracks depth identification. And Khatir et al.[10] proposed another hybrid model, the BOA-ANN algorithm optimizing ANN with Butterfly Optimization Algorithm (BOA), applied this model in identifying the crack depth in steel beam structures, and found that BOA-ANN can predict crack depth with improved accuracy. Furthermore, Achouri et al.[11] proposed a novel hybrid swarm optimization algorithm by combining PSO and BOA for the ANN optimization, adopted PSO-BOA-ANN for structural damage prediction, and found that PSO-BOA-ANN exhibits robust capability in structural damage detection.

The studies above show that machine learning can deliver remarkable benefits in solving complex engineering problems such as structural damage detection. Therefore, this paper also take advantage of machine learning to study the Anti-Penetration Randomness of Metal Protective Structures (APRMPS). And Back Propagation Neural Network (BPNN) based on the error back propagation algorithm is a well-recognized ANN with remarkably strong nonlinear mapping ability, and currently among the most widely used ANNs[12–14]. Nevertheless, the initial weights and thresholds of BPNN exert a significant impact on the training effects of the network, so if these parameters are not properly set, the training effects will be worse—a disadvantage of BPNN. On the other hand, GA is characterized by its strong global optimization ability, so it is effective to offset BPNN's disadvantage by adopting GA to optimize the initial weights and thresholds of BPNN. In fact, GA is often used as an optimization algorithm for BPNN[15–18]. What should be noted is that this optimization method is facing a challenge: the adoption of the selection operator for each generation of individuals in the population suggests too much human intervention in the process of population evolution, resulting in a low population evolution rate and thus reducing the efficiency of GA to optimize the initial weights and thresholds of BPNN. In order to address this challenge, this paper, based on the improvement of GA, proposes a new algorithm, Dynamic Lifecycle Genetic Algorithm (DLGA), which cancels the selection operation of GA and accordingly provides a life cycle for each individual in the population. Each individual automatically follows its development and reproduction within its own life cycle, so as to realize the natural evolution of the population, reducing the human intervention in the process of population evolution and enhancing the efficiency of optimizing the initial weights and thresholds of BPNN. Therefore, this paper investigates APRMPS by taking BPNN optimized by DLGA (DLGABPNN) as the surrogate model and adopting DLGABPNN-MCS, the Monte Carlo Simulation (MCS) with DLGABPNN calculation as repeated sampling tests.

In a nutshell, in exploring APRMPS by the method of DLGABPNN-MCS for the penetration probabilities of metal protective structures under the action of basic random variables, Section "APRMPS analysis" will analyze the possible candidates for basic random variables and random responses of APRMPS; Section "APRMPS study based on DLGABPNN-MCS" will propose DLGA, an improved version of GA, including its main processes of the optimization of BPNN, and the technical route of DLGABPNN-MCS to addressing APRMPS; Section "Applied examples", taking the anti-penetration randomness of two metal targets as an example, attempts to prove the higher efficiency of DLGA than GA in optimizing BPNN and verify the effectiveness of DLGABPNN-MCS in studying APRMPS. This paper hopes to promote the application of DLGA in other optimization problems and to provide reference for further study on APRMPS and study on other randomness problems.

APRMPS analysis
Basic random variables of APRMPS
In the numerical simulation of explosion and shock dynamics, for common metal materials, the principal stress is generally calculated by the Mie-Grüneisen equation of state[19]; the deviatoric stress by the Johnson–Cook plasticity model[19–21]; the damage degree by the Johnson–Cook failure model[21].

The Mie-Grüneisen equation of state[19] has two two expressions—the compression equation of state and the expansion equation of state

$$p = \begin{cases} \dfrac{\rho_0 c_0^2 \mu \left[1 + \mu\left(1 - \frac{\gamma_0}{2}\right) - \frac{1}{2}a\mu^2\right]}{\left[1 - \mu(S_1 - 1) - \frac{\mu^2 S_2}{\mu+1} - \frac{\mu^3 S_3}{(\mu+1)^2}\right]^2} + (\gamma_0 + a\mu)\, E, & (\mu > 0) \\ \rho_0 c_0^2 \mu + (\gamma_0 + a\mu)\, E, & (\mu < 0) \end{cases} \tag{1}$$

where p is the principal stress, $\mu = \frac{\rho}{\rho_0} - 1$ is the compression degree, with ρ_0 being the initial density and ρ the current density, c_0 is the sound velocity when $p = 0$, E is the energy in per unit volume, γ_0 is the Grüneisen coefficient when $p = 0$, and a, S_1, S_2 and S_3 are material parameters.

In the Johnson–Cook plasticity model[19–21], the equation for the Von Mises flow stress σ_y is

$$\sigma_y = \left(A + B\varepsilon_p^n\right)\left(1 + C\ln\dot{\varepsilon}_p^*\right)\left(1 - T^{*m}\right) \tag{2}$$

where ε_p is the equivalent plastic strain, $\dot{\varepsilon}_p^* = \frac{\dot{\varepsilon}_p}{\dot{\varepsilon}_0}$ is the dimensionless equivalent plastic strain rate, $T^* = \frac{T - T_r}{T_m - T_r}$ is the dimensionless homologous temperature, with T being the current temperature, T_r the room temperature and T_m the melting temperature, and A, B, C, n and m are material parameters.

In the Johnson–Cook failure model[21], the equation for the failure strain ε^f is

$$\varepsilon^f = [D_1 + D_2 \exp(D_3\sigma^*)]\left[1 + D_4 \ln \dot{\varepsilon}_p^*\right]\left[1 + D_5 T^*\right] \tag{3}$$

where $\sigma^* = p/\overline{\sigma}$ is the stress triaxiality, with $\overline{\sigma}$ being the equivalent stress, and D_1, D_2, D_3, D_4, and D_5 are material parameters.

From Eq. 1 to Eq. 3, γ_0, a, S_1, S_2 and S_3 in the Mie-Grüneisen equation of state, A, B, C, n and m in the Johnson–Cook plasticity model, and D_1, D_2, D_3, D_4, and D_5 in the Johnson–Cook failure model are material parameters. Jiang et al.[22] pointed out that even for the same material, the parameters given by different researchers may vary: for instance, Holmquist et al.[23] gave a different parameter value for the Johnson–Cook failure model of 4340 steel from the version proposed by Quan et al.[24], and Holmquist et al.[23] also gave a different parameter value for the Johnson–Cook plasticity model of 6061-t6 aluminum alloy from the version of Liu et al.[25]. Cullis et al.[26] and Normandia et al.[27] presented different parameter values for the Mie-Grüneisen equation of state of RHA steel. There are numerous examples for this, which shows that there is uncertainty in the determination of the parameter values for the performance equation of metal materials, leading to the uncertainty of the properties of the materials described by these parameters. What's more, in the study on the stochastic dynamics of metal structures under explosion load, Chen et al.[19,28] argued that each parameter for the Mie-Grüneisen equation of state, the Johnson–Cook plasticity model and the Johnson–Cook failure model are random, leading to the randomness of the principal stress, deviatoric stress and damage degree of metal structures under explosion load.

Moreover, errors in metal smelting and changes in environmental temperature will also cause changes in the density of metal materials, thus leading to the randomness of the metal material density. In the anti-penetration tests of metal protective structures, under the influence of the charge errors of the fragment launcher and the errors of the fragment velocity recording device, the penetration velocity of the fragment recorded in each test is also random.

Thus, in the study on APRMPS, since the randomness of the density and the performance equation parameters above of the materials of metal protective structures and fragments, and the randomness of the penetration velocity of fragments, may exert impacts on the penetration performance of fragments and the anti-penetration performance of metal protective structures, it is necessary to select from these parameters the ones that have a significant effect on the penetration results of metal protective structures as basic random variables.

Random response of APRMPS

In studying APRMPS, in order for ANN to identify the penetration results of metal protective structures under various randomness conditions, this paper introduces the penetration-recognizing coefficient δ to determine whether metal protective structures are penetrated under the action of basic random variables: if metal protective structures are penetrated, δ will be recognized to be 1; if not, δ will be recognized to be 0. This means that the randomness of the penetration of metal protective structures under basic random variables can be expressed by that of the 0–1 recognition results of δ. In this case, this paper defines δ as the random response of APRMPS, and considers APRMPS the problem of estimating the penetration probabilities of metal protective structures under basic random variables on the basis of the statistical analysis results of δ.

APRMPS study based on DLGABPNN-MCS
DLGA

(1) Life expectancy function and reproductive chromosome quantity function of individuals

In DLGA, an individual's life expectancy refers to the upper limit of its life. Being not a fixed value, the life expectancy needs to be continuously updated in iterative calculations. An individual's reproductive chromosomes refer to the chromosomes for the generation of new individuals through crossover operations and mutation operations. Being exactly the same as its own chromosome, the reproductive chromosomes can be generated by directly copying the latter. However, the quantity of an individual's reproductive chromosomes in each iteration is not a fixed neither. Therefore, this paper proposes the life expectancy function shown in Eq. 4 and the reproductive chromosome quantity function in Eq. 5 to determine in each iteration the number of life expectancy and reproductive chromosomes of each living individual in the population.

$$L_{ij} = \max\left\{\left\lceil C_1\left(1 - \frac{n_j}{N_L}\right)F_i\right\rceil, \ 1\right\} \tag{4}$$

where L_{ij} is the life expectancy of the individual i in the jth iteration calculation, F_i is the fitness of the individual i, n_j is the quantity of living individuals in the jth iteration calculation (i.e. excluding the unborn and dead individuals in the jth iteration calculation), N_L is upper limit quantity of living individuals, and C_1 is the coefficient of the equation.

$$Q_{ij} = \max\left\{\left\lceil C_2\left(1 - \frac{n_j}{N_L}\right)\left(1 - \frac{A_{ij}}{L_{ij}}\right)F_i\right\rceil, 0\right\} \tag{5}$$

where Q_{ij} is the quantity of the reproductive chromosomes generated by the individual i in the jth iteration calculation, A_{ij} is the age of the individual i in the jth iteration calculation, C_2 is the coefficient of the equation, and the other parameters are the same as those in Eq. 4.

(2) Crossover operator and mutation operator

The crossover operator and the mutation operator of DLGA are the same as those of GA. Thus in the real number coding mode, any two reproductive chromosomes can cross based on the specified crossover probability P_C, and the crossover operator can be expressed by Eq. 6:

$$\begin{cases} x_1' = (1-a)\,x_1 + ax_2 \\ x_2' = (1-a)\,x_2 + ax_1 \end{cases} \tag{6}$$

where x_1 and x_2 are respectively the genes crossed in any two reproductive chromosomes, x_1' and x_2' are respectively the newly-formed genes in these two reproductive chromosomes, and a, a random number within the interval [0, 1], represents the numeric exchange ratio of the gene x_1 to the gene x_2 in this crossover.

In the real number coding mode, each reproductive chromosome can mutate with the specified mutation probability P_M, and the mutation operator can be expressed by Eq. 7:

$$x' = \begin{cases} x + (x - x_{\max})\,f\,(N_j)\ , & r_1 > 0.5 \\ x + (x_{\min} - x)\,f\,(N_j)\ , & r_1 \le 0.5 \end{cases} \tag{7}$$

where x is the mutant gene in an reproductive chromosome, x' is the new gene generated after x mutated, $x_{\max}$ and $x_{\min}$ are respectively the upper limit and the lower limit to which x mutates, and r_1 is a random number within the interval [0, 1], which determines the mutation mode of x in this process—from Eq. 7 inferred two mutation modes. As shown in Eq. 8, $f(N_j)$ is a function with a range of [0, 1], which can guarantee that the variation of x in this mutation won't be excessively wide.

$$f\,(N_j) = r_2\,(1 - N_j / N_B)^2 \tag{8}$$

where N_j is the total quantity of having-been-born individuals while reaching the jth iteration calculation (the total quantity of the living and dead individuals while reaching the jth iteration calculation), N_B is the upper limit quantity of having-been-born individuals, and r_2 is a random number within the interval [0, 1], which can effectively reduce the value of the function $f(g)$.

(3) Life cycle of individuals

The life cycle of an individual can be so defined: before its birth, an individual's age is 0; after its birth, its initial age is 1, and its age is increased by 1 in each iteration calculation. Meanwhile, in each iteration calculation, an individual's life expectancy is calculated according to Eq. 4 and the quantity of its reproductive chromosomes is calculated according to Eq. 5, on which the replication of the individual's chromosome is based to become the reproductive chromosomes. The reproductive chromosomes will participate in the crossover operations and mutation operations to form new individuals. Once the individual's age reaches the life expectancy in an iterative calculation, the individual will die, which means that after that, the individual will no longer participate in the iterative calculation, that its age and life expectancy will no longer be updated, and that the individual will no longer generate reproductive chromosomes either.

(4) End conditions of DLGA

The 4 end conditions of DLGA in the jth iteration calculation are as follows:

Condition 1: No better individual appears in T_{end} consecutive iteration calculations, where T_{end} is the iterative calculation termination threshold that needs to be set artificially.
Condition 2: n_j drops to 0.
Condition 3: n_j reaches N_L.
Condition 4: N_j reaches N_B.

(5) Running process of DLGA

DLGA runs through the following process:
Step 1: Determine the fitness function. An individual's fitness function is determined by the practical situation and specific requirements of the optimization problem.
Step 2: Set parameters. Basic parameters should be set: ① n_1, the quantity of the first-generation individuals, ② N_L, ③ N_B, ④ T_{end}, ⑤ C_1, the coefficient of life expectancy function, ⑥ C_2, the coefficient of reproductive chromosome quantity function, ⑦ P_C, the crossover probabilities of reproductive chromosomes, and ⑧ P_M, the mutation probabilities of reproductive chromosomes.

Step 3: Generate first-generation individuals. The first-generation individuals of the population are generated in the same way as GA; the chromosome of each first-generation individual is encoded and the initial age of each first-generation individual is 1.

Step 4: Calculate each first-generation individual's fitness. The fitness function is applied to calculate each first-generation individual's fitness.

Step 5: Calculate the individuals' life expectancy. The life expectancy of each individual in the current iterative calculation is calculated according to Eq. 4.

Step 6: Form the population. Each individual's age and life expectancy are taken into account to determine whether they can continue to survive: If its age does not reach the life expectancy, the individual continues to survive; If its age has reached the life expectancy, it dies, and from then on, the iterative calculation will be terminated. Then, the individuals that continue to survive are aggregated into a population. The quantity of the living individuals and those of individuals in the previous generations in the population are counted.

Step 7: Generate reproductive chromosomes by individuals. The quantity of reproductive chromosomes to be generated by each living individual in each iteration is calculated according to Eq. 5, on which the replication of the individual's chromosome is based to become reproductive chromosomes.

Step 8: Update each individual's age. Each living individual's age is increased by 1.

Step 9: Generate new individuals. The crossover operations and mutation operations are performed on the reproductive chromosomes in the current iterative calculation in turn to generate new individuals, whose initial age is 1.

Step 10: Calculate new individuals' fitness. The fitness function is applied again to calculate each new individual's fitness.

Step 11: Determine whether the optimal individual is updated. If the fitness of each new individual is not higher than that of the current optimal individual, the optimal individual is not updated, and t_{end}, the number of consecutive iterations without better individuals, is increased by 1; otherwise, the new individual with the highest fitness is taken as the optimal, and t_{end} is reduced to 0.

Step 12: Determine whether the iteration calculation should continue. If none of the 4 end conditions of iteration calculation is met, optimization has not yet been completed, and Step 5-Step 12 should be repeated. If Condition 1 is met, or $t_{end} = T_{end}$, the global optimal point has successfully been searched and the iteration calculation will be terminated. If any of the other 3 conditions is met, the optimization fails, and it should return to Step 1 to start another round of optimization.

From the above fundamental principle of DLGA, it can be found that the essential difference between DLGA and GA lies in this fact: without the selection operation in GA, DLGA introduces parameters like an individual's age, life expectancy and reproductive chromosomes. An individual produces reproductive chromosomes during its survival, and once its age reaches life expectancy, it will die, thus forming its life cycle. Through the crossover operation and mutation operation in GA, each reproductive chromosome will develop a new individual, which starts its own life cycle. In this way, each individual in the population can automatically complete the replacement of the old with the new in its own life cycle, hence the natural evolution of the population. In short, DLGA can reduce human intervention in the process of population evolution, thereby enhancing the population evolution speed.

BPNN

BPNN is a typical multi-layer forward artificial neural network, and the following takes single-hidden-layer BPNN as an example to give a brief introduction to the principles of BPNN training[29].

The training of BPNN includes input information forward-propagation and error information back-propagation, and the process of the input information forward-propagation can be illustrated as follows:

It is assumed that the neuron quantities of the input layer, hidden layer and output layer of the BPNN are respectively n, m and l. Then, with the input of the training sample information into the BPNN, the output values of the neurons in the hidden layer and output layer of the BPNN can be respectively calculated by Eq. 9 and Eq. 10:

$$y_j = f\left(\sum_{i=1}^{n} v_{ij}x_i - \alpha_j\right) \tag{9}$$

$$o_k = f\left(\sum_{j=1}^{m} w_{jk}y_j - \beta_k\right) \tag{10}$$

where v_{ij} and w_{jk} are the weights of neurons, α_j and β_k are the thresholds of neurons, x_i is the input information, and y_j and o_k are respectively the output values of the neurons in the hidden layer and output layer. And the function f(net) is the activation function of neuron. If the f(net) is a bipolar Sigmoid function, it can be expressed by Eq. 11:

$$f(\text{net}) = \frac{1 - \exp(-\text{net})}{1 + \exp(-\text{net})} \tag{11}$$

The process of the error information back-propagation can be demonstrated as follows:

Firstly, the current iteration error information is calculated according to Eq. 12:

$$E = \frac{1}{2} \sum_{k=1}^{l} (d_k - o_k)^2 \tag{12}$$

where E is error information, d_k is the expected output value of each neuron in the output layer. Then, the corrected values of the weight and threshold of each neuron in the BPNN can be respectively calculated by Eq. 13–Eq. 16:

$$\Delta w_{jk} = \eta \left(d_k - o_k \right) o_k \left(1 - o_k \right) y_j \tag{13}$$

$$\Delta \beta_k = \eta \left(d_k - o_k \right) o_k \left(1 - o_k \right) \tag{14}$$

$$\Delta v_{ij} = \eta \left(\sum_{k=1}^{l} (d_k - o_k) o_k (1 - o_k) w_{jk} \right) y_j \left(1 - y_j \right) x_i \tag{15}$$

$$\Delta \alpha_j = \eta \left(\sum_{k=1}^{l} (d_k - o_k) o_k (1 - o_k) w_{jk} \right) y_j \left(1 - y_j \right) \tag{16}$$

where η is network training rate.

BPNN optimized by DLGA

Similar to the optimization of BPNN by GA[15–18], the optimization of BPNN by DLGA in this paper refers to the optimization of the initial weights and thresholds of BPNN by applying DLGA. Since each new individual generated by DLGA will give BPNN a complete set of initial weights and thresholds, and call BPNN for a complete training and testing process, the population evolution of DLGA is the optimization of the initial weights and thresholds of BPNN. When the genetic calculation is completed, the initial weights and thresholds corresponding to the optimal individuals are those of BPNN optimization, and the trained and tested BPNN corresponding to the optimal individuals is the optimized BPNN (DLGABPNN). The main process of the optimization of BPNN by DLGA is presented in Fig. 1 in the following.

APRMPS study based on DLGABPNN-MCS

DLGABPNN-MCS in this paper refers to the MCS that takes the DLGABPNN calculation as repeated sampling tests. The process of the APRMPS study based on DLGABPNN-MCS is presented in the following.

(1) Constructing DLGABPNN

First, the parameters that can exert a significant effect on the penetration results of metal protective structures are selected from the ones discussed in Section "Basic random variables of APRMPS" as basic random variables, and δ is taken as random response. After this, the sample points of the basic random variables are extracted and called in turn to perform repeated numerical simulations on APRMPS, so as to obtain the 0–1 recognition result of δ corresponding to each sample point. With the data being standardized, the standardized sample points of the basic random variables are used as the input data, and the 0–1 recognition results of δ are taken as the expected output data, thereby constructing the training sample sets and the test sample sets of DLGABPNN. Then, with the two types of sample sets, BPNN is optimized by applying DLGA, from which the optimal BPNN, or DLGABPNN, is developed for the APRMPS study.

(2) Applying DLGABPNN-MCS in APRMPS study

Firstly, the sample points are randomly selected based on the statistical distribution of basic random variables and are standardized to construct the prediction sample sets of APRMPS, which are later used for DLGABPNN-MCS, so as to quickly find the 0–1 recognition results of δ corresponding to the prediction sample sets. Finally, the 0–1 recognition results of δ are statistically analyzed to obtain the estimates of the penetration probabilities of metal protective structures.

The technical route shown in Fig. 2 summarizes the operation process of the APRMPS study with LCABPNN-MCS.

Applied examples
Example 1

This example adopted the anti-penetration tests of the metal targets reported in Reference 20, in which the targets were 12 mm-thick plane steel plates, and the fragments were tungsten alloy cubes of a size of 7.5 mm × 7.5 mm × 7.5 mm, with the measured result of $V_{50} = 821$ m/s and shear plugging.

Numerical model

In the repeated numerical simulations of the randomness study, this example took the generalized interpolation material point method (GIMPM)[30] for its numerical calculation, and applied a numerical calculation program based on a self-complied GIMPM program in the Fortran language, to which a self-developed material point free outflow boundary condition[31] was added, so that the fragments and the plugs could freely go beyond the
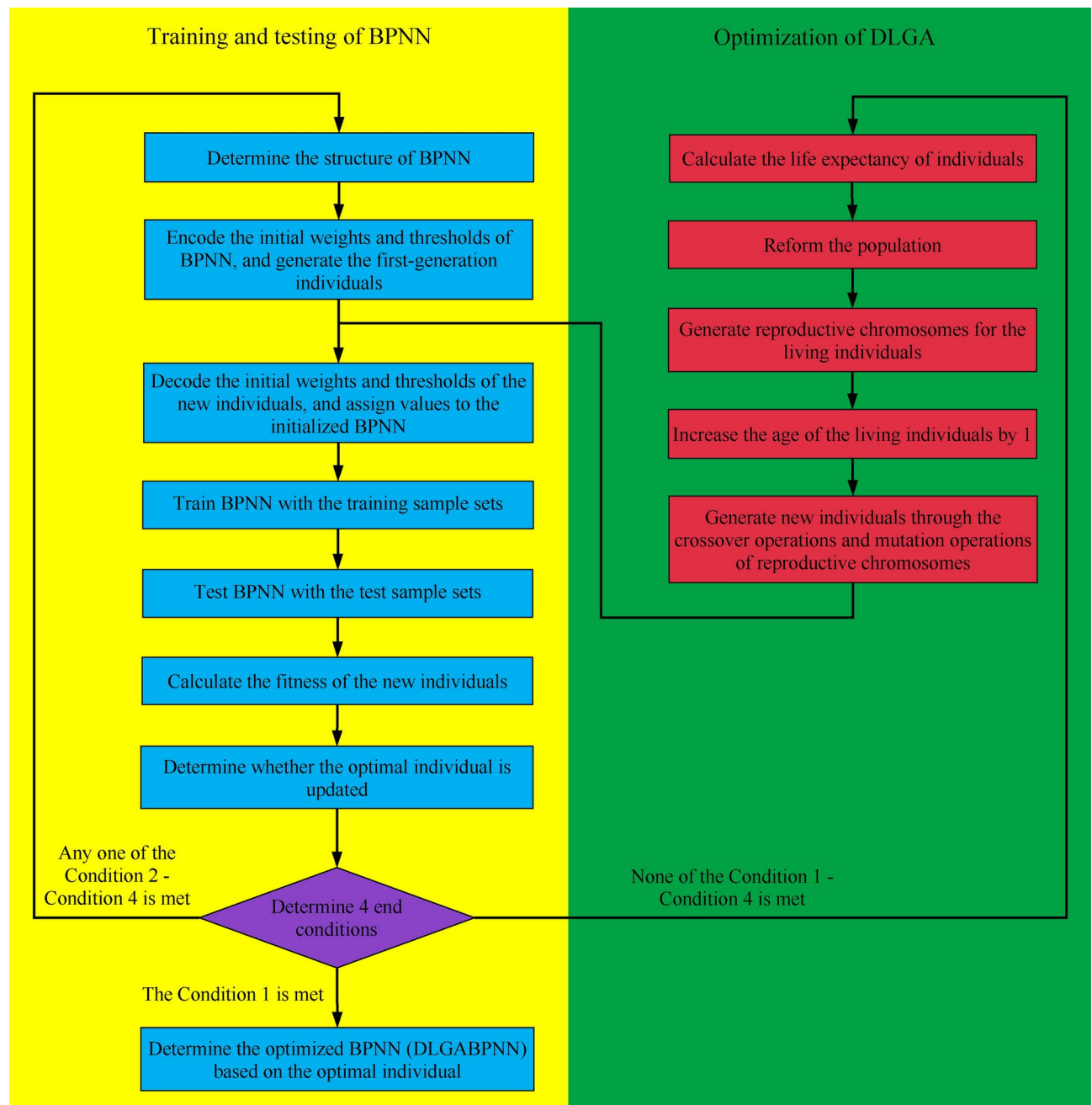
Fig. 1. Main process of the optimization of BPNN by DLGA.

background grid boundaries. In this way, a numerical model was constructed as shown in Fig. 3, and Tables 1, 2 and 3 show the deterministic parameter values for the Mie-Grüneisen equation of state, the Johnson–Cook plasticity model and the Johnson–Cook failure model of the steel and the tungsten alloy, with a density of 7.85 g/cm^3 and 17.3 g/cm^3 respectively.

For the accuracy of the basic random variables selected by the repeated numerical simulations of the GIMPM program, and that of the training sample set and the test sample sets of ANNs constructed in the same procedures, it is necessary to verify the accuracy of the GIMPM program to simulate the test results of this example in advance. Thus, the penetration velocity of the fragments was set at $V = V_{50} \pm 10$ m/s (i.e. 811 m/s and 831 m/s) and the numerical simulation results by the GIMPM program are shown in Fig. 4.

According to Fig. 4, when $V = 811$ m/s, the fragment failed to penetrate the target and was embedded in it, leaving a large depression of the target; when $V = 831$ m/s, the fragment successfully penetrated the target and caused shear plugging[20]. These simulation results are entirely consistent with the test results[20]. Therefore, the GIMPM program is capable of conducting accurate numerical simulations of this example, thus ensuring that the basic random variables selected, and the training sample set and the test sample sets of ANNs are accurate, meeting the demand from the APRMPS study.
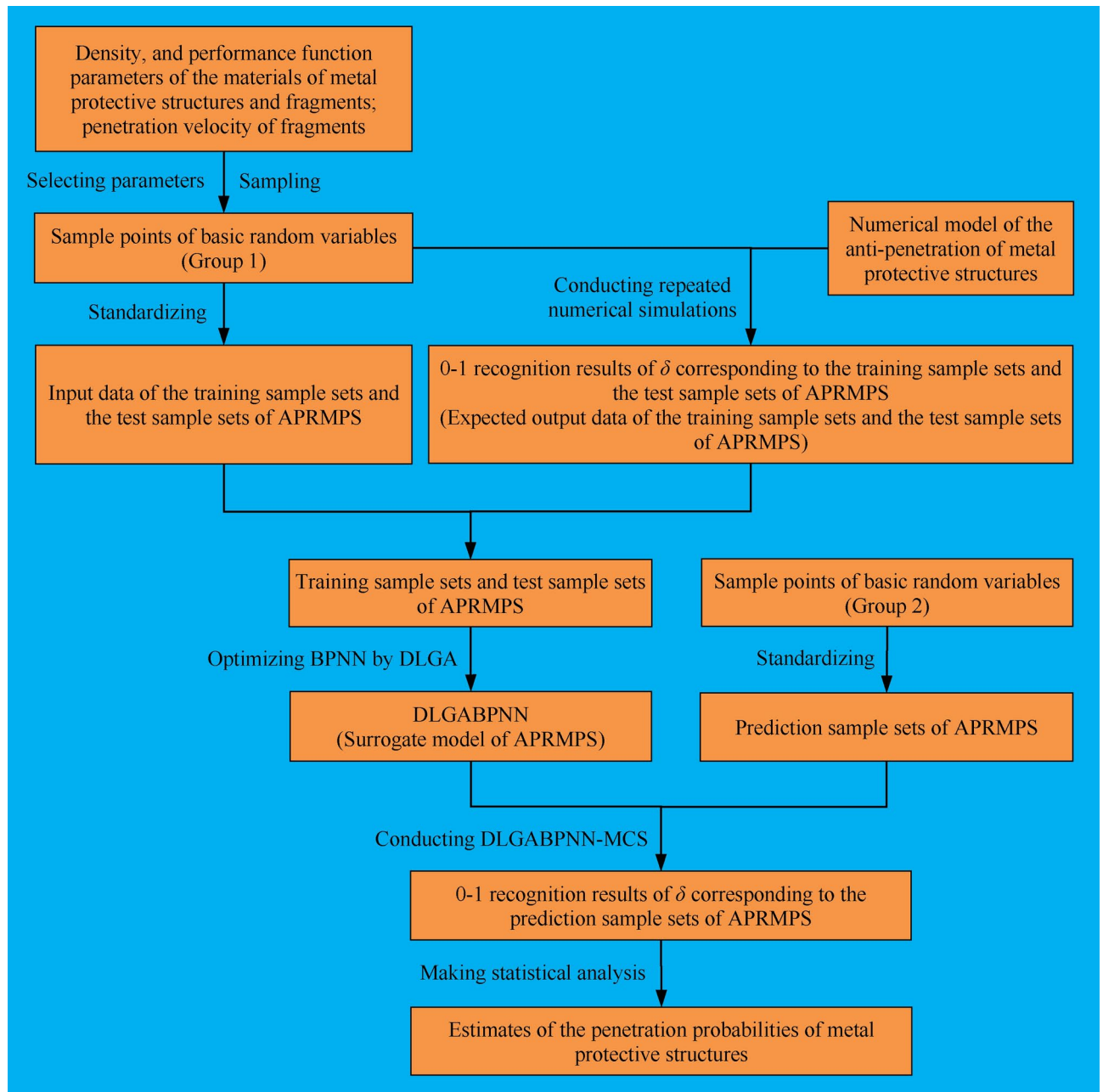
Fig. 2. Technical route to addressing APRMPS with DLGABPNN-MCS.

Selection of basic random variables

In order to make clear the parameters discussed in Section "Basic random variables of APRMPS", the APRMPS study labeled the density of, and the parameters in the Mie-Grüneisen equation of state, the Johnson–Cook plasticity model and the Johnson–Cook failure model of the fragment material (tungsten alloy) respectively as ρ_{F0}, γ_{F0}, a_P S_{F1}, S_{F2}, S_{F3}, A_P B_P C_P n_P m_P D_{F1}, D_{F2}, D_{F3}, D_{F4} and D_{F5}, and labeled the density of, and the parameters in the three equations/models of the target material (steel) respectively as ρ_{T0}, γ_{T0}, a_P S_{T1}, S_{T2}, S_{T3}, A_P B_P C_P n_P m_P D_{T1}, D_{T2}, D_{T3}, D_{T4} and D_{T5}. In order to reveal the common and universal aspects of the APRMPS study, it was assumed that the 33 parameters were all independent of each other and subject to the normal distribution. By changing the values of these 33 parameters one by one and repeating the numerical simulation shown in section "Numerical model", it was found that changing the values of the parameters V, ρ_{F0}, A_P ρ_{T0}, γ_{T0}, A_P B_P C_P n_P D_{T1} and D_{T2} could change the numerical simulation results. That is, when changing the values of these 11 parameters, it can lead the targets to be penetrated under the mean $\overline{V}$ of V is 811 m/s, and not be penetrated under $\overline{V}$ is 831 m/s. Thus it is justified to believe that these 11 parameters exert a significant influence on the penetration results of the targets in this study, and to take these 11 parameters as the basic random variables of the APRMPS in this example.
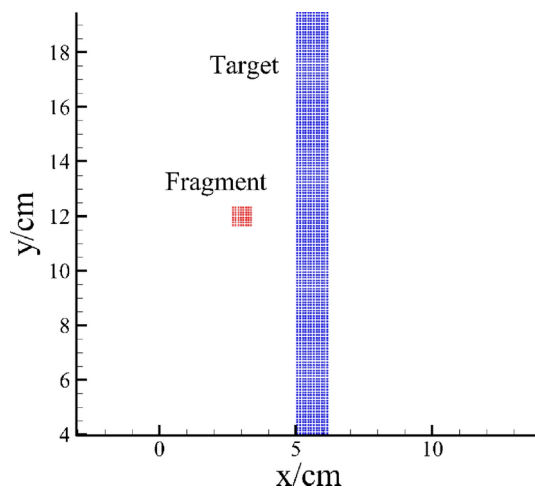
Fig. 3. Numerical model in Example 1.

Material	γ_0	S_1	S_2	S_3	a
Steel	2.17	1.49	0.0	0.0	0.0
Tungsten alloy	1.58	1.44	0.0	0.0	0.0

Table 1. Parameters for the Mie-Grüneisen equation of state in Example 1[32,33].

Material	A (MPa)	B (MPa)	n	C	m
Steel	1086.0	510.0	0.26	0.014	1.03
Tungsten alloy	1500.0	177.0	0.12	0.016	1.0

Table 2. Parameters for the Johnson–Cook plasticity model in Example 1[20].

Material	D_1	D_2	D_3	D_4	D_5
Steel	0.05	3.44	-2.12	0.002	1.61
Tungsten alloy	2.0	1.77	-3.4	0.0	0.0

Table 3. Parameters for the Johnson–Cook failure model in Example 1[28,34].

APRMPS study

In order to reveal the common and universal aspects of the APRMPS study, it was assumed that the 11 basic random variables were all independent of each other and subject to the normal distribution, so the means of these basic random variables were their deterministic parameter values as shown in Tables 1, 2 and 3. Then, DLGABPNN-MCS was adopted to study APRMPS under the 4 randomness conditions that the values of the coefficients of variation (CVs) of these basic random variables were respectively 0.01, 0.03, 0.05 and 0.07. Additionally, in the ANN training and testing phases of this example, BPNN and the BPNN optimized by GA (GABPNN) worked as the performance comparison objects of DLGABPNN in order to explore the advantages of DLGABPNN as a surrogate model for numerical models.

(1) Constructing the training sample set and the test sample set

For the generalization capabilities of DLGABPNN, GABPNN and BPNN, under the randomness conditions that the CVs of basic random variables were all 0.1 and that $\overline{V} = V_{50}$ (821 m/s), 2000 sample points were selected by applying the Latin Hypercube Sampling, and then underwent the repeated numerical simulations based on the GIMPM program discussed in Section "Numerical model" to obtain the 0–1 recognition results of δ corresponding to these sample points, so as to construct a 2000-point training sample set for training the 3 ANNs. Further, in the interval of $V_{50} \pm 100$ m/s, being gradually increased by 10 m/s each time, each value of $\overline{V}$ worked as the center, and under the 4 randomness conditions that the values of the CVs of basic random variables were respectively 0.01, 0.03, 0.05 and 0.07, 20 sample points were randomly selected for each value of $\overline{V}$, on which the repeated numerical simulations were conducted for the 0–1 recognition results of δ. Thus, the
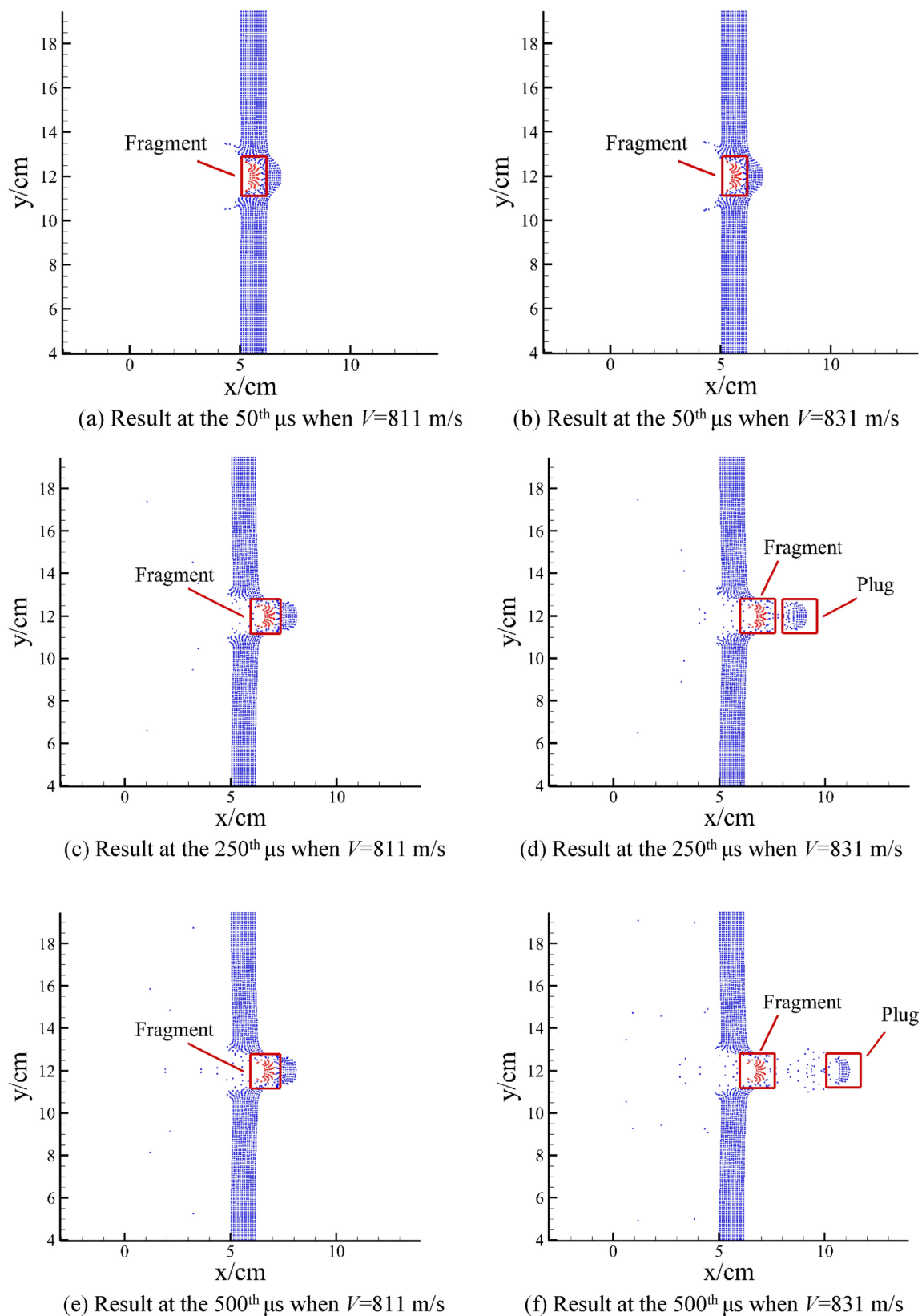
Fig. 4. Numerical simulation results when $V = 811$ m/s, and $V = 831$ m/s.

1680-point test sample set was constructed in order to test the accuracy of the 3 ANNs in predicting the 0–1 recognition results of δ.

(2)   Determining the fitness function

This example took Eq. 17 as the fitness function of DLGABPNN and GABPNN.

$$F_i = C_3 R_i + C_4 \tag{17}$$

where F_i is the fitness of the individual i, R_i is the test accuracy of ANN corresponding to the individual i, both C_3 and C_4 are the coefficients of this equation, and $C_3 > 0$.

In Eq. 17, F_i is proportional to R_i. It ensures that DLGA and GA can optimize BPNN while improving the test accuracy of the BPNN, and thus obtain the surrogate model that can accurately predict the penetration results of the targets.

(3)   Determining the structure of and key parameters for ANNs

Since Cotter[35] and Castro et al.[36] have proven that the 3-layer forward ANNs can solve the majority of the nonlinear mapping problems, the DLGABPNN in this example adopted the single hidden-layer ANNs. Besides, with 11 basic random variables and an random response (δ) in this example, the quantity of the input-layer neurons of DLGABPNN was set to 11; that of the output-layer neurons 1; that of the hidden-layer neurons 15, thus constructing an DLGABPNN structure of $11 \times 15 \times 1$. Then, the maximum number of network training times T_{max} was set to 20,000, the network training rate S_t 0.05, the allowable error E_a of network training 0.001, n_1 10, N_L 500, N_B 10,000, T_{end} 5, P_C 0.8, P_M 0.1, and the coefficients C_1, C_2, C_3 and C_4 in Eq. 4, Eq. 5 and Eq. 17 were 0.2, 0.3, 120.0 and -100.0, respectively.

GABPNN had the same structure as DLGABPNN, with the population size M_p being 50 and the maximum reproduction generation G_{max} being 60. The rest of the parameters were the same as their counterparts in DLGABPNN. And BPNN also had the same structure and parameters as DLGABPNN.

(4)   Performance comparison between DLGA and GA for BPNN optimization

Given the stochastic nature of DLGA and GA in optimizing BPNN, a rigorous comparative analysis was conducted for assessing their performances. Specifically, each was used to optimize BPNN in 200 random and independent trials. Also, BPNN was trained and tested 200 times randomly and independently as a comparison. The test accuracy values and their statistical characteristics of the 3 ANNs are shown in Fig. 5 and Table 4, and the serial numbers and statistical characteristics of the optimal individuals in the DLGA and GA populations are presented in Fig. 6 and Table 5.

As shown in Fig. 5 and Table 4, the 200 DLGABPNNs and 200 GABPNNs have very close mean test accuracy, both of which are significantly higher than the mean test accuracy of the 200 BPNNs. Also, the standard deviations of the test accuracy of DLGABPNNs and GABPNNs are similar and markedly lower than those of BPNNs. This means that DLGA and GA have comparable optimization effects on BPNN, and that each optimization yields the best-performing BPNN, with test accuracy stably around 0.98. In contrast, unoptimized BPNNs show substantially lower accuracy and greater variability, reflecting inherent instability and sub-optimal generalization.
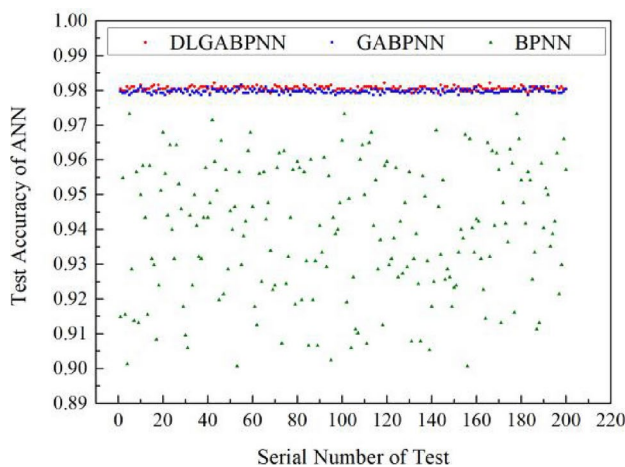
Fig. 5. Comparison of the test accuracies of the ANNs in Example 1.

Statistical characteristic of test accuracy	Mean	Standard deviation
DLGABPNN	0.9805	5.9706×10^{-4}
GABPNN	0.9797	5.8186×10^{-4}
BPNN	0.9392	1.8620×10^{-2}

Table 4. The statistical characteristics of the test accuracy of the ANNs in Example 1.
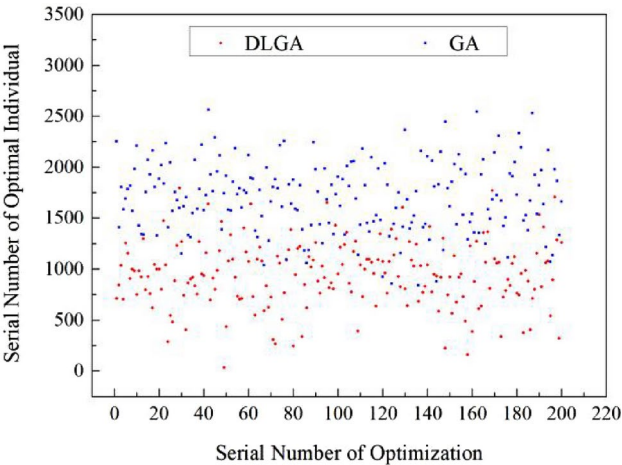
Fig. 6. Comparison of the serial numbers of optimal individuals in Example 1.

Statistical characteristic of serial numbers of optimal individuals	Mean	Standard deviation
DLGA	942.7750	319.2522
GA	1702.1801	346.5127

Table 5. The statistical characteristics of the serial numbers of optimal individuals in Example 1.

Figure 6 and Table 5 indicate that the mean serial number of the 200 optimal individuals in DLGA populations is smaller than that in GA populations, while the standard deviation of these serial numbers for DLGA is comparable to that of GA. This suggests that DLGA generally achieves higher optimization efficiency than GA for BPNN, while the stability of its optimization efficiency remains statistically comparable to that of GA.

The comparative analysis of DLGA and GA in optimizing BPNN reveals the following key findings: both DLGA and GA demonstrate comparable performance in optimizing BPNN, consistently yielding optimal BPNN capable of accurately predicting the 0–1 recognition results of δ for APRMPS; however, DLGA exhibits superior optimization efficiency over GA. Thus, it is more reasonable to use DLGA in optimizing BPNN for the construction the surrogate model of APRMPS, and subsequent application of DLGABPNN-MCS reliably estimates the P's of targets, meeting the accuracy requirements.

(5) Estimating the P's of the targets based on DLGABPNN-MCS

First, in the interval of $V_{50} \pm 300$ m/s, being gradually increased by 10 m/s each time, each value of $\overline{V}$ worked as the center, and under the 4 randomness conditions discussed above, 10,000 sample points were randomly selected for each value of $\overline{V}$ to construct four 610,000-point prediction sample sets. Then, DLGABPNN-MCS was employed to obtain the 0–1 recognition results of δ of the 4 prediction sample sets, so as to obtain the estimates of the P's of the targets corresponding to each value of $\overline{V}$ under the 4 randomness conditions. Finally, 4 relation curves between P and $\overline{V}$ (the P-$\overline{V}$ curves) are drawn as shown in Fig. 7, on which the data in Table 6 are based.

Three observations can be obtained from Fig. 7 and Table 6. First, the $\overline{V}_{50}$ of the targets calculated by DLGABPNN-MCS under the above 4 randomness conditions is consistently 819 m/s, nearly the same with the V_{50} (821 m/s) measured in the test[20]. This proves that DLGABPNN-MCS was perfectly accurate in estimating the P's of the targets. Second, the P-$\overline{V}$ curves are monotonically increasing curves that converge to 0% and 100% respectively at the left and right ends. This shows that under the action of basic random variables, when $\overline{V} \le \overline{V}_0$, $P = 0\%$, and the targets will definitely not be penetrated, that when $\overline{V} \ge \overline{V}_{100}$, $P = 100\%$, and the targets will certainly be penetrated, and that when $\overline{V}_0 < \overline{V} < \overline{V}_{100}$, whether the targets can be penetrated is random,
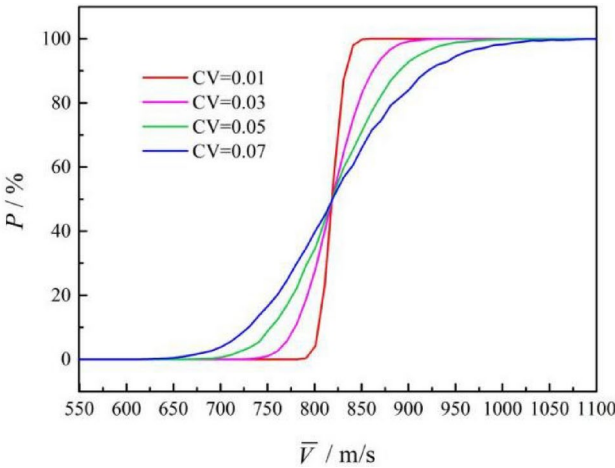
Fig. 7. P-$\overline{V}$ curves in Example 1.

Randomness condition	$\overline{V}_0$ (m/s)	$\overline{V}_{50}$ (m/s)	$\overline{V}_{100}$ (m/s)
CV = 0.01	792	819	849
CV = 0.03	741	819	911
CV = 0.05	696	819	975
CV = 0.07	652	819	1048

Table 6. $\overline{V}_0$, $\overline{V}_{50}$ and $\overline{V}_{100}$ of the targets under the 4 randomness conditions in Example 1.

with P increasing with $\overline{V}$. Last, the larger the CVs of basic random variables, the stronger the randomness of the APRMPS, and the smaller the value of $\overline{V}_0$ and the larger the value of $\overline{V}_{100}$, which leads to a longer increasing interval of $[\overline{V}_0, \overline{V}_{100}]$ of the P–$\overline{V}$ curves, where the curves increase more steadily.

Example 2

This example employed the anti-penetration tests of the metal targets reported in Reference 21, in which the targets were 15 mm-thick plane aluminum alloy plates, and the fragments were steel cubes of a size of 10 mm × 10 mm × 8 mm, with the measured result of $V_{50} = 595.3$ m/s and shear plugging.

Numerical model
For the repeated numerical simulations of the randomness study, this example adopted the same numerical simulation method and program as Example 1, and constructed a numerical model shown in Fig. 8. In addition, the steel had a density of 7.85 g/cm³ and the aluminum alloy 2.78 g/cm³. The deterministic parameter values for the Mie–Grüneisen equation of state, the Johnson–Cook plasticity model and the Johnson–Cook failure model of the latter are shown in Tables 7, 8 and 9, and the deterministic parameter values of the three performance equations of steel are the same as those in Example 1 (See Tables 1, 2 and 3).

In order to verify the accuracy of the test results simulated by the GIMPM program, the penetration velocity of the fragments was set at $V = V_{50} \pm 10$ m/s (i.e. 585.3 m/s and 605.3 m/s) and the numerical simulation results by the GIMPM program are shown in Fig. 9.

It can be seen from Fig. 9 that when $V = 585.3$ m/s, the fragment failed to penetrate the target, leaving a large depression of the target, and that when $V = 605.3$ m/s, the fragment successfully penetrated the target and caused shear plugging. These results are fully consistent with the test results[21]. Therefore, the GIMPM program is qualified to conduct accurate numerical simulations of this example.

Selection of basic random variables
The density of, and the parameters in the Mie-Grüneisen equation of state, the Johnson–Cook plasticity model and the Johnson–Cook failure model of the fragment material (steel) were labeled respectively as $\rho_{F0}, \gamma_{F0}, a_P S_{F1}, S_{F2}, S_{F3}, A_P B_P C_P n_P m_P D_{F1}, D_{F2}, D_{F3}, D_{F4}$ and D_{F5}, and the density of, and the parameters in the three performance equations of the target material (aluminum alloy) were marked respectively as $\rho_{T0}, \gamma_{T0}, a_P S_{T1}, S_{T2}, S_{T3}, A_P B_P C_P n_P m_P D_{T1}, D_{T2}, D_{T3}, D_{T4}$ and D_{T5}. It was also assumed that the 33 parameters were all independent of each other and subject to the normal distribution. By changing the values of these 33 parameters one by one and repeating the numerical simulation shown in section "Numerical model", it was found that changing the values of the parameters $V, \rho_{F0}, \rho_{T0}, A_T B_T n_T D_{T1}$ and D_{T2} could change the numerical simulation results. That is, when changing the values of these 8 parameters, it can lead the targets to be penetrated under the mean $\overline{V}$ of V is 585.3 m/s, and not be penetrated under $\overline{V}$ is 605.3 m/s. Thus it proves that these 8 parameters
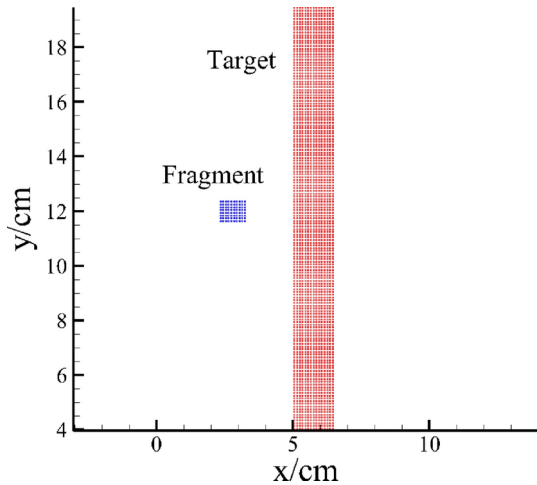
Fig. 8. Numerical model in Example 2.

Material	γ_0	S_1	S_2	S_3	a
Aluminum alloy	2.0	1.338	0.0	0.0	0.0

Table 7. Parameters for the Mie–Grüneisen equation of state in Example 2[37].

Material	A (MPa)	B (MPa)	n	C	m
Aluminum alloy	369.0	684.0	0.73	0.0083	1.7

Table 8. Parameters for the Johnson–Cook plasticity model in Example 2[37].

Material	D_1	D_2	D_3	D_4	D_5
Aluminum alloy	0.13	0.13	-1.5	0.011	0.0

Table 9. Parameters for the Johnson–Cook failure model in Example 2[37].

exert a significant influence on the penetration results of the targets in this study, so these 8 parameters were taken as the basic random variables of the APRMPS in this example.

APRMPS study
It was also assumed that the 8 basic random variables were all independent of each other and subject to the normal distribution, so the means of these basic random variables were their deterministic parameter values. And DLGABPNN-MCS was also employed to study the APRMPS under the 4 randomness conditions that the values of the CVs of these basic random variables were respectively 0.02, 0.04, 0.06 and 0.08. Like Example 1, in the ANN training and testing phases, BPNN and GABPNN worked as the performance comparison objects of DLGABPNN in order to discuss the advantages of DLGABPNN as a surrogate model for numerical models.

(1) Constructing the training sample set and the test sample sets

First, under the randomness conditions that the CVs of basic random variables were all 0.1 and that $\overline{V} = V_{50}$ (595.3 m/s), 2000 sample points were withdrawn by applying the Latin Hypercube Sampling, and then underwent the repeated numerical simulations for the 0–1 recognition results of δ, so as to construct a 2000-point training sample set for training the 3 ANNs. Then, in the interval of $V_{50} \pm 100$ m/s, being gradually increased by 10 m/s each time, each value of $\overline{V}$ worked as the center, and under the 4 randomness conditions that the values of the CVs of basic random variables were respectively 0.02, 0.04, 0.06 and 0.08, 20 sample points were randomly selected for each value of $\overline{V}$, on which the repeated numerical simulations were conducted for the 0–1 recognition results of δ. Thus, the 1680-point test sample set was constructed in order to test the accuracy of the 3 ANNs in predicting the 0–1 recognition results of δ.
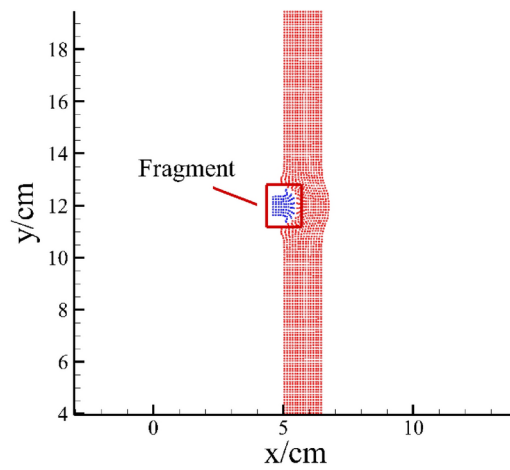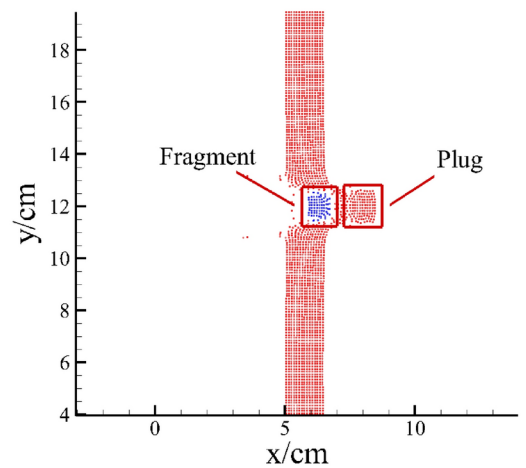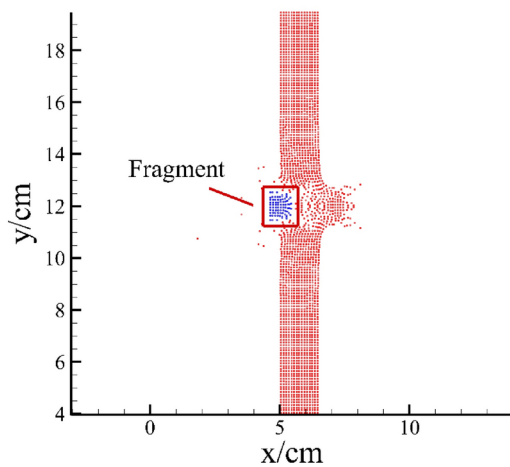
(a) Result at the 50th μs when V=585.3 m/s
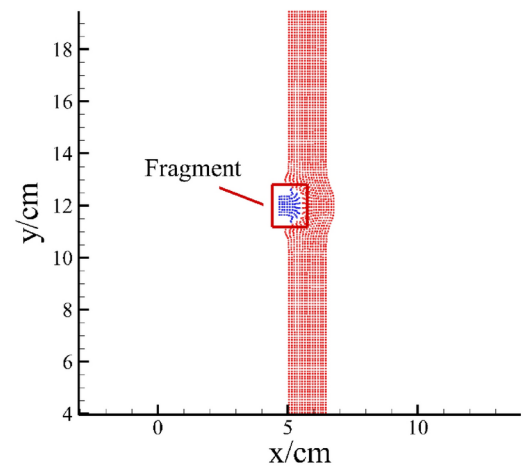
(b) Result at the 50th μs when V=605.3 m/s
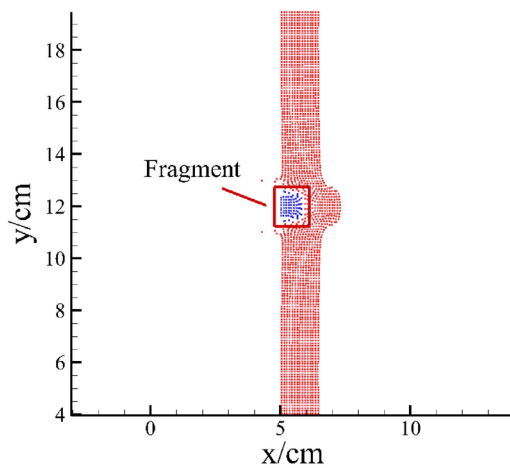
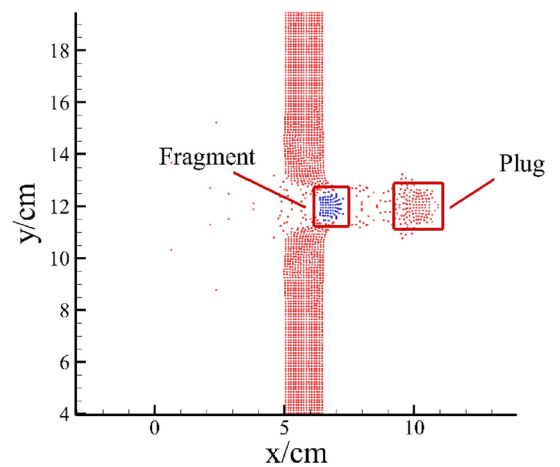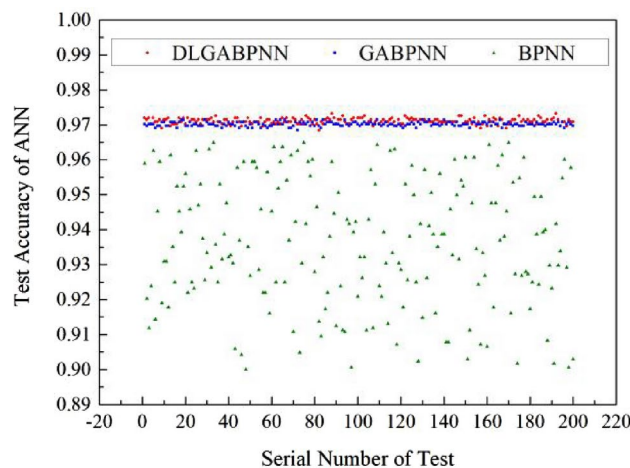(c) Result at the 250th μs when V=585.3 m/s

(d) Result at the 250th μs when V=605.3 m/s

(e) Result at the 500th μs when V=585.3 m/s

(f) Result at the 500th μs when V=605.3 m/s

Fig. 9. Numerical simulation results when $V = 585.3$ m/s, and $V = 605.3$ m/s.

Fig. 10. Comparison of the test accuracy of the ANNs in Example 2.

Statistical characteristic of test accuracy	Mean	Standard Deviation
DLGABPNN	0.9711	8.2253×10^{-4}
GABPNN	0.9702	6.3276×10^{-4}
BPNN	0.9362	1.8001×10^{-2}

Table 10. The statistical characteristics of the test accuracy of the ANNs in Example 2.

(2) Determining the fitness function

This example also took Eq. 17 as the fitness function of DLGABPNN and GABPNN.

(3) Determining the structure of and key parameters for ANNs

DLGABPNN, GABPNN and BPNN in this example adopted the single hidden-layer ANNs. Besides, with 8 basic random variables and an random response (δ) in this example, the quantities of the input-layer neurons of the 3 ANNs were set to 8; those of the output-layer neurons 1; those of the hidden-layer neurons 15, thus constructing the 3 ANNs with a structure of $8 \times 15 \times 1$. Then, the parameters of the ANNs in this example were the same as their counterparts in Example 1.

(4) Performance comparison between DLGA and GA for BPNN optimization

A comparative analysis of DLGA and GA in optimizing BPNN was also conducted for assessing their performances. Similarly, each was used to optimize BPNN in 200 random and independent trials. Also, BPNN was trained and tested 200 times randomly and independently as a comparison. The test accuracy values and their statistical characteristics of the 3 ANNs are shown in Fig. 10 and Table 10, and the serial numbers and statistical characteristics of the optimal individuals in the DLGA and GA populations are presented in Fig. 11 and Table 11.

From Fig. 10 and Table 10, it can be seen that the 200 DLGABPNNs and 200 GABPNNs have very close mean test accuracy, both of which are significantly higher than the mean test accuracy of the 200 BPNNs. Also, the standard deviations of the test accuracy of DLGABPNNs and GABPNNs are similar and markedly lower than those of BPNNs. This means that DLGA and GA have comparable optimization effects on BPNN, and that each optimization yields the best-performing BPNN, with test accuracy stably around 0.97. In contrast, unoptimized BPNNs show substantially lower accuracy and greater variability, resulting in low and highly random test accuracy each time.

Figure 11 and Table 11 show that the mean of the serial numbers of the 200 optimal individuals of DLGA is smaller than that of GA, and the standard deviation of these serial numbers of the optimal individuals of DLGA is also smaller than that of GA. This means that DLGA generally achieves higher and more stable optimization efficiency than GA for BPNN.

The comparison above indicates that both DLGA and GA demonstrate comparable performance in optimizing BPNN, consistently yielding optimal BPNN capable of accurately predicting the 0–1 recognition results of δ for APRMPS, with DLGA exhibiting superior optimization efficiency over GA. Therefore, it is more reasonable
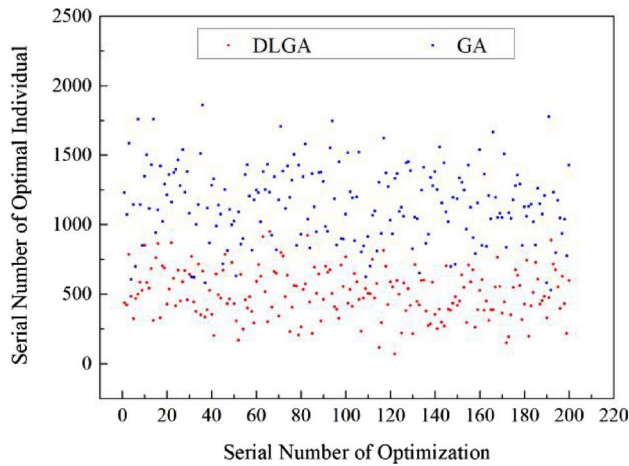
Fig. 11. Comparison of the serial numbers of optimal individuals in Example 2.

Statistical characteristic of serial numbers of optimal individuals	Mean	Standard deviation
DLGA	503.4100	174.5481
GA	1150.6200	267.6031

Table 11. The statistical characteristics of the serial numbers of optimal individuals in Example 2.
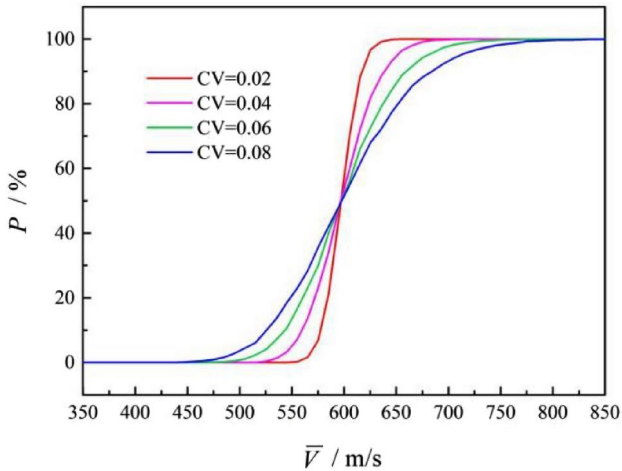
Fig. 12. P-$\overline{V}$ curves in Example 2.

to use DLGA in optimizing BPNN for the construction the surrogate model of APRMPS, and subsequent application of DLGABPNN-MCS reliably estimates the P's of targets, meeting the accuracy requirements.

(5) Estimating the P's of the targets based on DLGABPNN-MCS

First, in the interval of $V_{50} \pm 300$ m/s, being gradually increased by 10 m/s each time, each value of $\overline{V}$ worked as the center, and under the 4 randomness conditions discussed above, 10,000 sample points are randomly selected for each value of $\overline{V}$ to construct four 610,000-point prediction sample sets. Then, DLGABPNN-MCS was employed to obtain the 0–1 recognition results of δ of the 4 prediction sample sets, so as to obtain the estimates of the P's of the targets corresponding to each value of $\overline{V}$ under the 4 randomness conditions. Finally, 4 P-$\overline{V}$ curves are drawn in Fig. 12, on which the data in Table 12 are based.

Figure 12 and Table 12 lead to three conclusions similar to those of Fig. 7 and Table 6 in Example 1. First, the $\overline{V}_{50}$ of the targets calculated by DLGABPNN-MCS under the above 4 randomness conditions is consistently 597 m/s, nearly the same with the V_{50} (595.3 m/s) measured in the test[21], which proves that DLGABPNN-MCS

Randomness condition	$\overline{V}_0$ (m/s)	$\overline{V}_{50}$ (m/s)	$\overline{V}_{100}$ (m/s)
CV = 0.02	557	597	640
CV = 0.04	526	597	682
CV = 0.06	495	597	737
CV = 0.08	465	597	794

Table 12. $\overline{V}_0$, $\overline{V}_{50}$ and $\overline{V}_{100}$ of the targets under the 4 randomness conditions in Example 2.

was perfectly accurate in estimating the P's of the targets. Second, the P–$\overline{V}$ curves are monotonically increasing curves that converge to 0% and 100% respectively at both ends. The last conclusion is that the larger CVs of basic random variables the longer increasing interval of $[\overline{V}_0, \overline{V}_{100}]$ of the P–$\overline{V}$ curves.

Discussion

This paper contributes 2 primary innovations. First, it analyzes APRMPS and systematically investigates its potential candidates for basic random variables, introducing δ as the random response of APRMPS. Second, it proposes DLGA by improving GA, and presents the main process of optimizing BPNN using DLGA and the technical route to addressing APRMPS with DLGABPNN-MCS. By applying this method, the penetration probabilities of metal protective structures under the action of basic random variables can be obtained.

However, 3 aspects of this paper still need improvement in future study. First, the randomness of parameters in the Mie-Grüneisen equation of state, the Johnson–Cook plasticity model, and the Johnson–Cook failure model for metallic materials requires extensive experimental validation. Future studies may conduct systematic experiments by employing Bayesian sampling to quantify statistical distributions of these parameters for the three equations and to investigate their impact on the stochasticity of metallic material dynamics. Second, this paper focuses on the anti-penetration randomness of single metallic structures under fragment impact, while composite material structures are also commonly used in armor protection in engineering. Thus, it is necessary to further investigate the anti-penetration randomness of composite material structures in future research. Third, this paper proposes DLGA based on the improvement of GA and applies it to optimize BPNN for the surrogate model, while emerging artificial intelligence and machine learning offer promising alternatives, as suggested in References[5–11]. Therefore, future studies will explore other state-of-the-art ANNs as surrogate models to investigate engineering problems such as the anti-penetration randomness of protective structures and structural health monitoring.

Conclusion

It is necessary to extend the anti-penetration of metal protective structures to APRMPS and to apply ANN-MCS in the APRMPS study for the penetration probabilities of metal protective structures under the action of basic random variables. From the density of, and the parameters for the Mie-Grüneisen equation of state, the Johnson–Cook plasticity model and the Johnson–Cook failure model of the materials of the metal protective structures and the fragments, this method also selects the parameters exerting significant effects on the penetration results of metal protective structures as the basic random variables of APRMPS. It takes δ, which represents whether metal protective structures are penetrated, as the random response of APRMPS.

Moreover, DLGA proposed in this paper removes the selection operation and constructs the life cycles of individuals, in which each individual automatically follows its development and reproduction, so as to reduce the human intervention and ensure the natural evolution of the population, hence a higher optimization efficiency of DLGA, compared to GA. Also, this paper presents the main processes of the optimization of BPNN by DLGA and the technical route of DLGABPNN-MCS to addressing APRMPS, demonstrates the specific procedures for this method in the two applied examples of the anti-penetration randomness of metal targets, proves the higher efficiency of DLGA than GA in optimizing BPNN, and verifies the effectiveness of DLGABPNN-MCS in studying APRMPS.

In addition, Like GA, DLGA is also a universal heuristic algorithm for resolving various optimization problems, including optimizing ANNs, it is advisable to extend its application in follow-up study for the exploration of other optimization problems. Besides, since this is the debut of DLGA, there is still much room for the improvement of the technical details, such as life expectancy function, reproductive chromosome quantity function and specific procedures of DLGA.

Furthermore, future studies should conduct in-depth research into the randomness of parameters in the Mie–Grüneisen equation of state, the Johnson–Cook plasticity model and the Johnson–Cook failure model to uncover their statistical patterns and impacts on the stochastic dynamic behavior of these materials. It is also essential to explore the anti-penetration randomness of composite material protective structures and other complex structures. Additionally, more advanced ANNs should be employed as surrogate models to investigate engineering problems such as the anti-penetration randomness of protective structures and structural health monitoring.

Data availability

The datasets used and analysed during the current study available from the corresponding authors on reasonable request.

References

1. Sevkat, E. Experimental and numerical approaches for estimating ballistic limit velocities of woven composite beams. *Int. J. Impact Eng.* **45**, 16–27. https://doi.org/10.1016/j.ijimpeng.2012.01.007 (2012).
2. Li, Y. et al. Effects of water on the ballistic performance of para-aramid fabrics: Three different projectiles. *Text. Res. J.* **86**(13), 1372–1384. https://doi.org/10.1177/0040517515612355 (2016).
3. Ali, A. et al. Ballistic impact properties of woven bamboo-woven E-glass-unsaturated polyester hybrid composites. *Def. Technol.* **15**(3), 282–294. https://doi.org/10.1016/j.dt.2018.09.001 (2019).
4. Cho, C. et al. Evaluation of ballistic limit velocity using instrumented indentation test of 7xxx aluminum alloys after friction stir welding. *Met. Mater. Int.* **27**(7), 2264–2275. https://doi.org/10.1007/s12540-020-00680-2 (2021).
5. Khatir, A. et al. Multiple damage detection and localization in beam-like and complex structures using co-ordinate modal assurance criterion combined with firefly and genetic algorithms. *J. Vibroeng.* **20**(1), 832–842. https://doi.org/10.21595/jve.2016.19719 (2016).
6. Khatir, A. et al. An efficient improved gradient boosting for strain prediction in near-surface mounted fiber-reinforced polymer strengthened reinforced concrete beam. *Front. Struct. Civ. Eng.* **18**(8), 1148–1168. https://doi.org/10.1007/s11709-024-1079-x (2024).
7. Khatir, A. et al. Advancing structural integrity prediction with optimized neural network and vibration analysis. *J. Struct. Integr. Maint.* **9**(3), 2390258. https://doi.org/10.1080/24705314.2024.2390258 (2024).
8. Khatir, A. et al. Enhancing damage detection using reptile search algorithm optimized neural network and frequency response function. *J. Vib. Eng. Technol.* **13**, 88. https://doi.org/10.1007/s42417-024-01545-3 (2025).
9. Khatir, A. et al. A new hybrid PSO-YUKI for double cracks identification in CFRP cantilever beam. *Compos. Struct.* **311**, 116803. https://doi.org/10.1016/j.compstruct.2023.116803 (2023).
10. Khatir, A. et al. Vibration-based crack prediction on a beam model using hybrid butterfly optimization algorithm with artificial neural network. *Front. Struct. Civ. Eng.* **16**(8), 976–989. https://doi.org/10.1007/s11709-022-0840-2 (2022).
11. Achouri, F. et al. Structural health monitoring of beam model based on swarm intelligence-based algorithms and neural networks employing FR. *J. Braz. Soc. Mech. Sci. Eng.* **45**, 621. https://doi.org/10.1007/s40430-023-04525-y (2023).
12. Wang, Y. L., Liu, Y. & Che, S. B. Research on pattern recognition based on BP neural network. *Adv. Mater. Res.* **282–283**, 161–164. https://doi.org/10.4028/www.scientific.net/AMR.282-283.161 (2011).
13. Xue, Z. C. & Wang, H. J. The application of artificial BP neural networks and Monte-Carlo method for the reliability analysis on frame structure. *Appl. Mech. Mater.* **204–208**, 3256–3259. https://doi.org/10.4028/www.scientific.net/AMM.204-208.3256 (2012).
14. Yang, A. M. et al. Design of intrusion detection system for internet of things based on improved BP neural network. *IEEE Access* **7**, 106043–106052. https://doi.org/10.1109/ACCESS.2019.2929919 (2019).
15. Zhao CJ. Research of Expression Recognition Base on Optimized BP Neural Network. In *16th International Conference on Industrial Engineering and Engineering Management*, China, Beijing vol. **1–2** 1803–1806. https://doi.org/10.1109/ICIEEM.2009.5344321 (2009).
16. Zheng, B. H. Material procedure quality forecast based on genetic BP neural network. *Mod. Phys. Lett. B* **31**(19–21), 1740080. https://doi.org/10.1142/S0217984917400802 (2017).
17. Li, H. X. et al. Application of BP neural network based on genetic algorithm optimization. In *4th International Conference on Intelligent Information Processing*, China, Guilin 160–165. https://doi.org/10.1145/3378065.3378096 (2019).
18. Liu, L. et al. Study on shock initiation randomness of energetic materials on a macroscopic scale. *Appl. Sci.* **13**(4), 2534. https://doi.org/10.3390/app13042534 (2023).
19. Chen, W. D. et al. Stochastic material point method for analysis in non-linear dynamics of metals. *Metals* **9**(1), 107. https://doi.org/10.3390/met9010107 (2019).
20. Cheng, Y. et al. Influence of dimensionless parameters on ballistic limit velocity of fragments. *J. Ballist.* **34**(4), 8–14. https://doi.org/10.12115/j.issn.1004-499X(2022)04-002 (2022).
21. Xin, T. & Han, Q. Research on ballistic limit velocity of steel fragments penetrating target. *Sci. Technol. Eng.* **12**(2), 264–268. https://doi.org/10.3969/j.issn.1671-1815.2012.02.004 (2012).
22. Jiang, D. et al. Penetration of confined AD95 ceramic composite targets by tungsten long rods. *Explos. Shock Waves* **30**(1), 91–95. https://doi.org/10.3969/j.issn.1001-1455.2010.01.015 (2010).
23. Holmquist, T. J., Templeton, D. W. & Bishnoi, K. D. Constitutive modeling of aluminum nitride for large strain, high-strain rate, and high-pressure applications. *Int. J. Impact. Eng.* **25**(3), 211–231. https://doi.org/10.1016/S0734-743X(00)00046-4 (2001).
24. Quan, X. et al. Numerical simulation of long rods impacting silicon carbide targets using JH-1 model. *Int. J. Impact. Eng.* **33**(1–12), 634–644. https://doi.org/10.1016/j.ijimpeng.2006.09.011 (2006).
25. Liu, Z. Q. et al. State-of-the-art of constitutive equations in metal cutting operations. *Tool Eng.* **42**(3), 3–9. https://doi.org/10.16567/j.cnki.1000-7008.2008.03.018 (2008).
26. Cullis, I. G. & Lynch, N. J. Performance of model scale lone rod projectiles against complex targets over the velocity range 1700–2200 m/s. *Int. J. Impact. Eng.* **17**, 263–274. https://doi.org/10.1016/0734-743X(95)99852-I (1995).
27. Normandia, M. J. & Lee, M. Penetration performance of multiple segmented rods at 26 km/s. *Int. J. Impact. Eng.* **23**, 675–686. https://doi.org/10.1016/S0734-743X(99)00113-X (1999).
28. Chen, W. D. et al. A stochastic material point method for probabilistic dynamics and reliability. *Comput. Mech.* **63**(5), 1069–1082. https://doi.org/10.1007/s00466-018-1667-5 (2019).
29. Han, L. Q. & Shi, Y. *Artificial Neural Network Theory and Application* 1st edn, 42–76 (China Machine Press, 2018).
30. Bardenhagen, S. G. & Kober, E. M. The generalized interpolation material point method. *CMES Comput. Model. Eng. Sci.* **5**, 477–495. https://doi.org/10.1007/s11831-004-0004-3 (2004).
31. Chen, W. D. et al. Improvement of the material point method and its application in explosion and impact dynamics. *J. Harbin Eng. Univ.* **42**(11), 1671–1678. https://doi.org/10.11990/jheu.202008057 (2021).
32. Zhang, S. W. et al. Measuring and numerical simulation of attenuation of planar shock wave in PMMA. *Acta Armamentarii* **37**(7), 1214–1219. https://doi.org/10.3969/j.issn.1000-1093.2016.07.008 (2016).
33. Xu, Y. X., Ren, J. & Wang, S. S. Research on perforation limit thickness of low carbon steel plates impacted normally by tungsten spheres. *Trans. Beijing Inst. Technol.* **37**(6), 551–556. https://doi.org/10.15918/j.tbit1001-0645.2017.06.001 (2017).
34. Cheng, Y. et al. A numerical simulation study of typical cube fragments invading armored steel. *J. Ordnance Equip. Eng.* **43**(8), 106–111. https://doi.org/10.11809/bqzbgcxb2022.08.016 (2022).
35. Cotter, N. E. The stone-weierstrass theorem and its application to neural networks. *IEEE Trans. Neural Netw.* **1**, 290–295. https://doi.org/10.1109/72.80265 (1990).
36. Castro, J. L., Mantas, C. J. & Benitez, J. M. Neural networks with a continuous squashing function in the output are universal approximators. *Neural Netw.* **13**, 561–563. https://doi.org/10.1016/S0893-6080(00)00031-9 (2000).
37. Buyuk, M. et al. Automated design of threats and shields under hypervelocity impacts by using successive optimization methodology. *Int. J. Impact Eng.* **35**(12), 1449–1458. https://doi.org/10.1016/j.ijimpeng.2008.07.057 (2008).

Author contributions

Conceptualization, L.L. and W.C.; methodology, L.L., W.C., and S.L.; software, Y.Y.; validation, L.L. and Y.Y.; formal analysis, L.L.; investigation, L.L., S.L., and M.S.; data curation, L.L., S.L., and M.S.; writing—original draft preparation, L.L.; writing—review and editing, W.C. and S.L.; supervision, W.C. and S.L. All authors have read and agreed to the published version of the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to L.L. or S.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.