

New hybrid between SPEA/R with deep neural network: Application to predicting the multi-objective optimization of the stiffness parameter for powertrain mount systems

Journal of Low Frequency Noise,
Vibration and Active Control
0(0) 1–16
© The Author(s) 2019
DOI: 10.1177/1461348419868322
journals.sagepub.com/home/lfn



Dinh-Nam Dao^{1,2}  and Li-Xin Guo¹

Abstract

In this study, a new methodology, hybrid Strength Pareto Evolutionary Algorithm Reference Direction (SPEA/R) with Deep Neural Network (HDNN&SPEA/R), has been developed to achieve cost optimization of stiffness parameter for powertrain mount systems. This problem is formalized as a multi-objective optimization problem involving six optimization objectives: mean square acceleration of a rear engine mount, mean square displacement of a rear engine mount, mean square acceleration of a front left engine mount, mean square displacement of a front left engine mount, mean square acceleration of a front right engine mount, and mean square displacement of a front right engine mount. A hybrid HDNN&SPEA/R is proposed with the integration of genetic algorithm, deep neural network, and a Strength Pareto evolutionary algorithm based on reference direction for multi-objective SPEA/R. Several benchmark functions are tested, and results reveal that the HDNN&SPEA/R is more efficient than the typical deep neural network. Stiffness parameter for powertrain mount systems optimization with HDNN&SPEA/R is simulated, respectively. It proved the potential of the HDNN&SPEA/R for stiffness parameter for powertrain mount systems optimization problem.

Keywords

SPEA/R algorithm, feed-forward artificial neural network, extreme learning machine, multi-objective evolutionary algorithms, powertrain mount system stiffness, mounting system

Introduction

Multi-objective evolutionary algorithms (MOEAs) are common tools for solving multi-objective optimization problems in the technical field, because of their performance on issues with large design spaces and scenes difficult exercise. Inside, SPEA2 (Intensity 2 Evolutionary Algorithm) is used to evaluate the Pareto solution due to the good performance of a variety of solutions different from normal multi-objective reliability assessment. There are some researchers on this field like De Tommasi et al.¹ who have proposed a multi-objective optimization for RF circuit blocks through replacement models and NBI and SPEA2 methods. Sofianopoulos and Tambouratzis² proposed a model-based machine translation system using large monolithic blocks in the target language from which statistical information was extracted. This study reported using a specific machine translation to represent the test that SPEA2 was chosen as the optimization method. Zhao et al.³ proposed an SPEA2 algorithm based on adaptive selection evolutionary operators (AOSPEA). The proposed algorithm can selectively adapt simulated

¹School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China

²Control Technology College, Le Quy Don Technical University, Hanoi, Vietnam

Corresponding author:

Dinh-Nam Dao, No 3-11, Wenhua Road, Heping District, Shenyang 110819, China.

Email: daodinhnambk@gmail.com



binary interference, polynomial mutations, and differential evolutionary operators in their evolution according to their contribution to the external repository. Ben Hamidabrini Salah et al.⁴ proposed the Pareto Strength (SPEA2) Evolution Algorithm for the Economic/Environmental Power Distribution (EPPD) problem. In the past, minimizing fuel costs was the only objective function of economic power coordination. Due to the modification of clean air behavior has been applied to reduce emissions of polluting emissions from power plants, utilities have also changed strategies to reduce pollution and atmospheric emissions, minimizing generation waste when other target functions turn economic capacity (EPD) into versatile-objective problem with conflicting goals. Jiang et al.⁵ published a new SPEA based on the reference direction, denoted SPEA/R, to optimize multiple goals. A significant extension of the early SPEA algorithms is SPEA/R. It applies to the advantage of SPEA2's physical assignment in quantifying solutions diversity and convergence in one method. It is appropriate to replace the most time-consuming density estimator with an algorithm based on the reference direction. Their proposed exercise duties also take into account the convergence both local and global. However, MOEAs algorithms still need a lot of computational time to evaluate the objective function in the typical practical problem-solving process. When the problem is more difficult and complicated, the calculation time will be longer. Combined, this could make the use of MOEAs algorithms impractical. Therefore, the best way to reduce computation time is to use artificial neural networks (ANNs) with multiple hidden layers plus deep learning algorithms to accelerate calculations.

The ANNs with hidden layers combined with deep learning algorithms is one of the most widely used and accurate predictive models. Many researchers have applied this method in the areas such as economics, engineering, society, foreign exchange, securities issues, etc.⁶⁻¹⁴ The application of neural networks in predictive models optimizes many goals based on the ability of neural networks to predict that non-fixed behavior is very accurate. For traditional mathematical models or statistical models, it is inconsistent with unusual data patterns that cannot be clearly written as functions or deduced from a formula, while ANN algorithms can work with chaotic components. Currently, there are a number of researchers such as Shen et al.,¹⁵ who have come up with potential uncertainties of wind power and have since proposed options for building intervals prediction (PI) with predictive models using wavelet neural network, in which the upper and lower limit estimate of PI has been implemented by minimizing the multi-objective function including the probability of span width and coverage range. Smith et al.¹⁶ announced a recurrent neural network used as an alternative method to predict long-term models of fluid dynamics simulation in computation. In particular, hybrid MOEAs have been trained and optimized structures from introduced recurrent neural networks. Kakaee et al.¹⁷ published a method of using ANNs followed by multi-objective optimization using NSGA-II evolution algorithm and SPEA2 optimization algorithm to optimize the operating parameters of a compression ignition heavy-duty diesel engine. Vieira and Tome¹⁸ published two different methods to increase the search speed of the multi-objective evolution algorithm (MOEA) using ANNs.

In this article, a new hybrid optimization algorithm is proposed for multi-objective problems. This is the hybrid between the genetic algorithm (GA), deep neural network (DNN), and strength Pareto evolutionary algorithm-based reference direction for multi-objective (SPEA/R) to find the best of the Pareto-optimal front. This combination gave computing time much faster than computing time when using GAs SPEA/R. On the other hand, this combination also significantly reduces the number of samples needed for the training of deep ANNs. The performance of the new algorithm is demonstrated via some complex benchmark functions and for powertrain mount system stiffness parameter optimization problem with six-objective optimization in model 3D.

The organization of this article is as follows: section "structure" describes the proposed new hybrid HDNN&SPEA/R method and the vibration characteristic of the powertrain mount system. Section "simulation results and discussion" describes simulation results of application HDNN&SPEA/R method to computational experimentation with several benchmark functions and optimization of the powertrain mount system stiffness parameter. Finally is a conclusion.

Structure

Many-objective optimization SPEA/R algorithm

Jiang et al. proposed SPEA/R algorithm as presented in the flowchart in Figure 1.

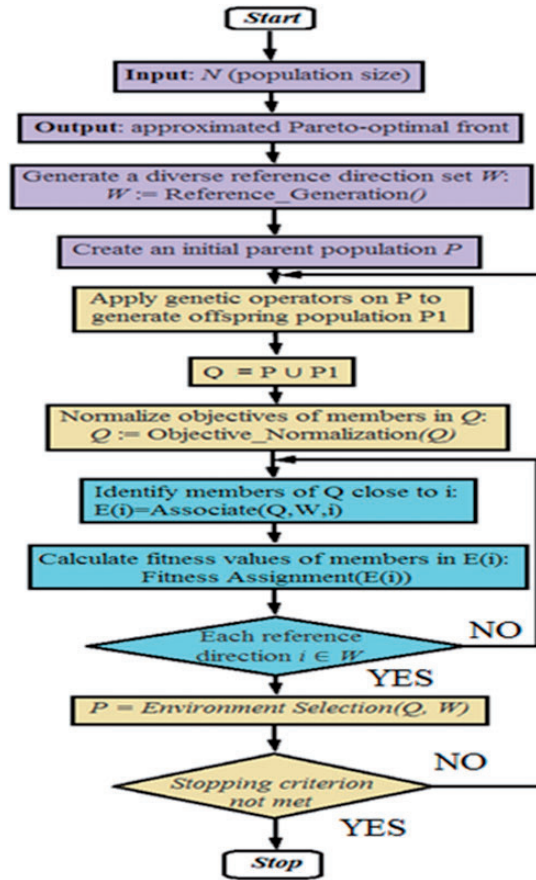


Figure 1. Flowchart of SPEA/R.

W: reference direction set; P: initial parent population; P1: offspring population; Q: summarize the population of parents and children; E(i): fitness values function.

New hybrid HDNN&SPEA/R method

DNN. ANNs are simulations of simplified models of the human brain. They have the ability to estimate complex nonlinear relationships between corresponding input and output data parameters.¹⁹ As shown in Figure 2, the neural network has a link structure consisting of three types of classes. It is an input layer, hidden layer, and output layer. Each network layer consists of a number of neurons and is organized into layers, in which the nerve cells of different layers in the network are connected by connections connected to independent weights (W_i). In addition, an independent bias (b) can be added to each neuron. On the other hand, the transfer function determines the influence of the weights and biases of the neuron on the neurons of the next layer and can be linear or nonlinear. In addition, there are several types of transfer functions (such as pureline, logig, and tansig); some neurons and some hidden layers are hyperparameters of neural networks to create different structures of the network. neuron. Finally, weights and biases adjustment process are called the network training process, and they are usually evaluated by minimizing the average mean square error (MSE) between the predicted outputs of the neural network and the output reality.

In this article, we use the multilayer perceptron (MLP) neural network structure. MLP is a feedforward ANN defined by an input layer with M neurons. N layers hidden, in which each hidden layer has N_h number of neurons and an output layer has K neurons. In the MLP network structure, each layer has full connections to the next layer, which means that each neuron output in layer N is the input of each neuron in the $N + 1$ layer. Figure 2 shows one example of an MLP network with input neuron M , N hidden layer with each hidden layer has N_h neuron, and the output layer has K neurons. MLP network can be described with: $nn = (M N_{h1} N_{h2} \dots N_{hN} K)$.

DNN optimize. Recently, some researchers have published some new effective methods to train ANN neural networks, in which the weights and biases of neural networks are optimized by GA algorithm.²⁰ In the process

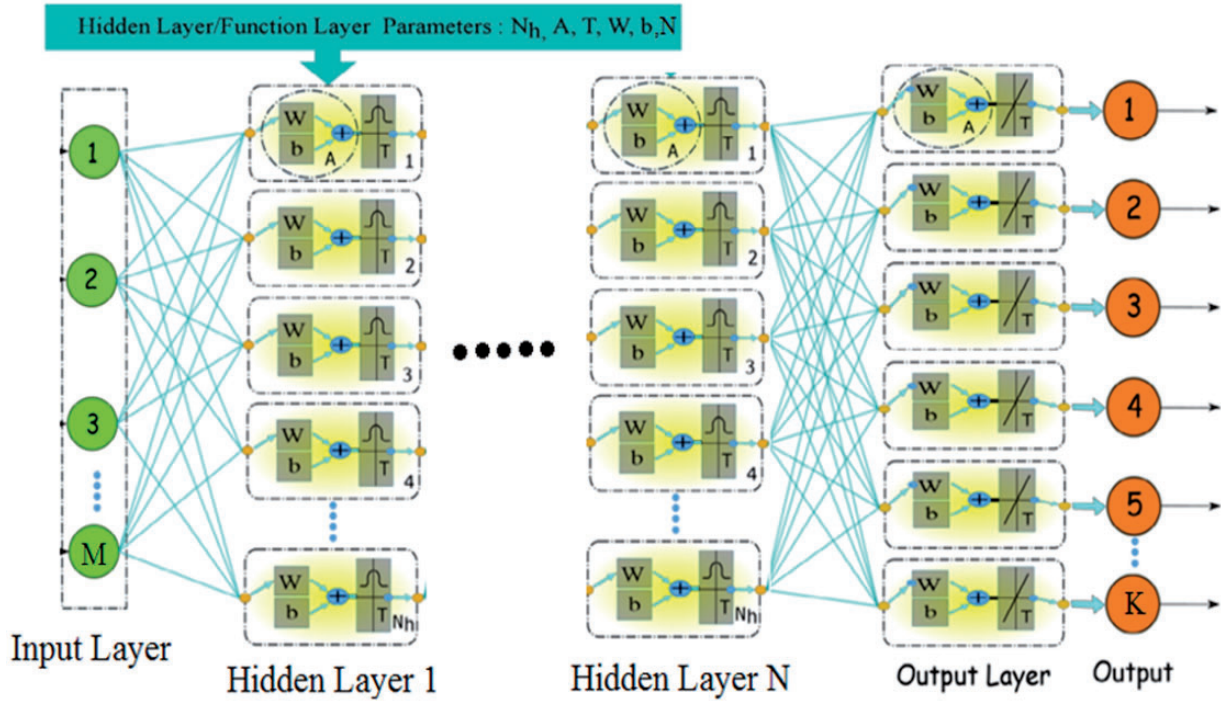


Figure 2. The architecture of a feedforward artificial neural network with M neurons in input, N in hidden layer, and K in output layers.

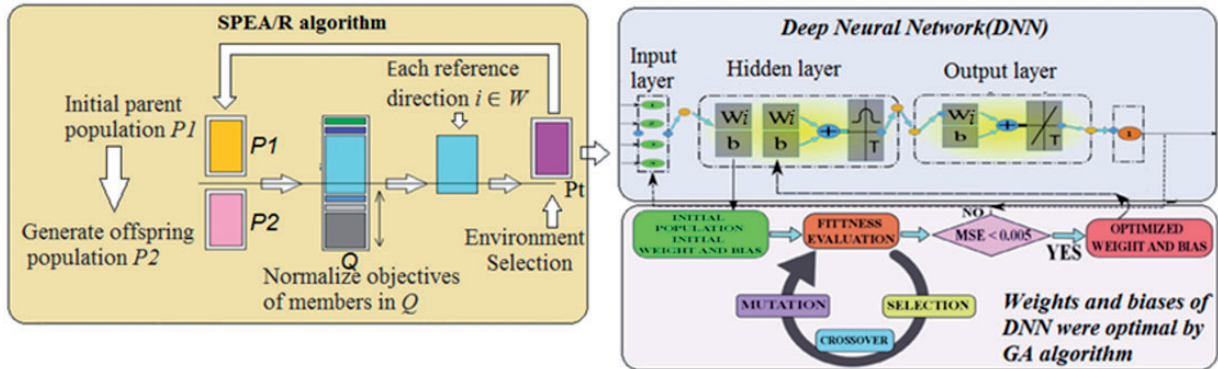


Figure 3. Training structure for hybrid method.

of network training, GA finds weight and bias values quickly and optimally for neural networks. That makes the number of iterations in network training greatly reduced. Thus, the time for training is faster. Besides, the global ability of searching and evolution of parameters is a key feature of GA. Therefore, GA was introduced in this study. This method uses the theory of natural selection and biological evolution. It is the choice, cross-exchange, and mutation of individuals to select the best and most suitable member. The initial weight and bias of DNN have been evolved in the process of training neural networks. The interaction between the GA and DNN is done through weight and bias exchange. The DNN was started to get a random weight and biases (W_i and b) as shown in Figure 3. This is the initial population included in the GA algorithm. Then, the next generation is generated by the GA based on the current population. To evaluate the difference between the predicted output values and the actual output values, it is used as the fitness function. Decide on acceptable parameters if the total average square of GA is less than 0.005. Weight and biases are calculated by the equation (1)

$$N_w = (I_n + 1)N_h + (N_h + 1)O_p \quad (1)$$

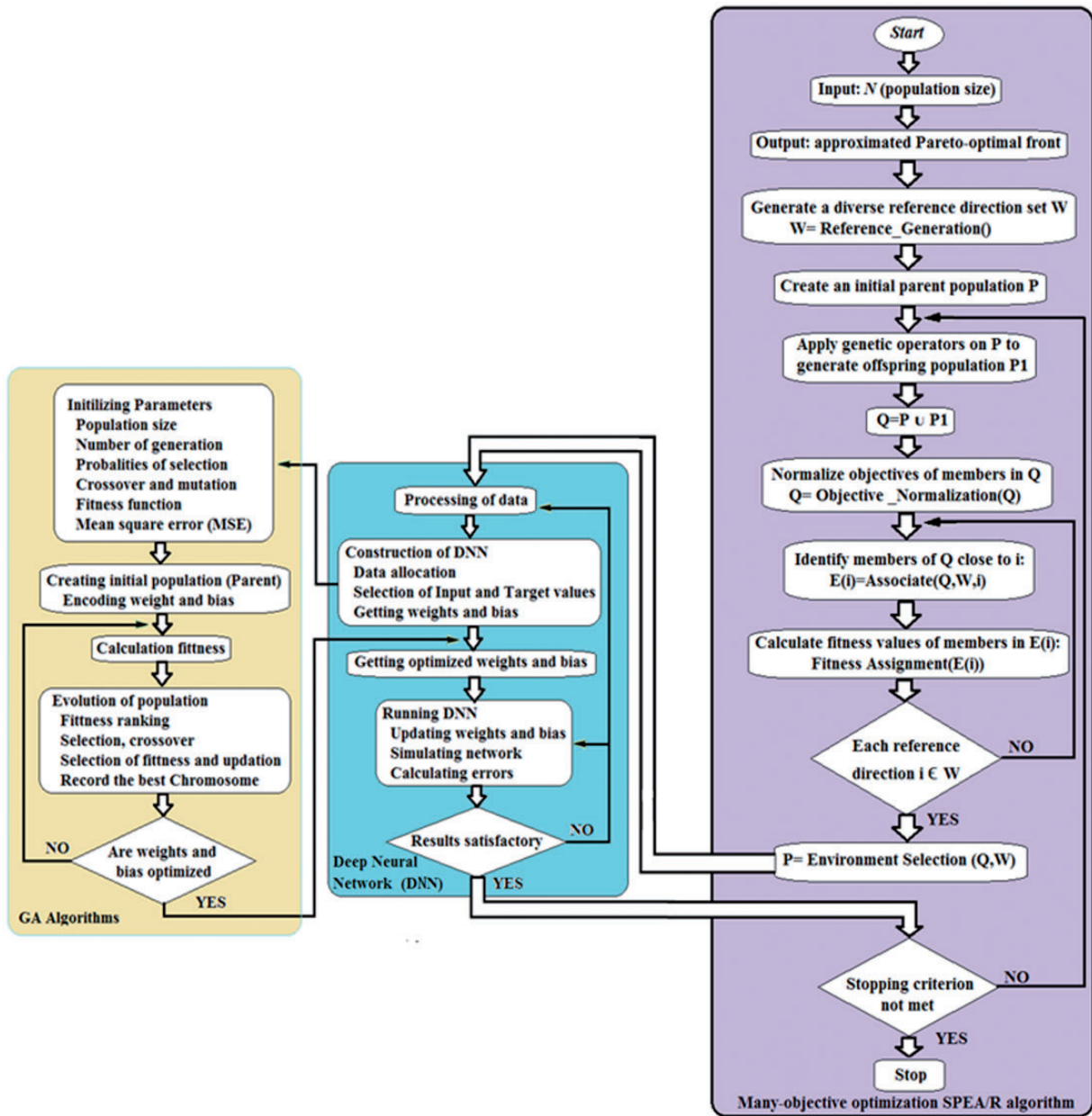


Figure 4. Training algorithm.

where N_w is a number of weights and deviations, I_n is a number of neurons in the input layer, N_h is some hidden neurons, and an O_p number of neurons in the output layer. Through the training process, the optimal GA value has been achieved. GA with population size 20, the mutation rate 0.15, and the crossover rate 0.65 has been chosen for GA operation. GA has been run for 250 generations.

The deep learning training algorithm. The purpose of this section is to allow the DNN neural network to learn the optimal analytical characteristics of many objects from the SPEA/R algorithm. We have combined DNN training with the optimal analysis of SPEA/R algorithm as shown in Figures 3 and 4. This process started with the SPEA/R algorithm with a random population of input variables P_1 . After that, summing up the population Q at time t is created by population P_1 and population P_2 (where P_2 is generated from P_1 parents' population through the use of conventional genetic operators such as selection, mutations, and cross exchanges). After that, all individuals were combined into population Q .

From this population, individuals were selected to enhance reproduction instead of random selection. It is therefore very useful for optimizing many goals when remote parents are unable to create good solutions. The use of simple normalization based on the worst value of each SPEA/R goal deliberately gives higher priority to diversity than convergence when making environmental choices leading to printing performance for MOP issues. From there, the best Pareto fronts were selected (stored on the top of the list) and transferred to the new parent group P_t . Through this process, the most quintessential nuclei were selected. Since the size of the P_t population is only half of Q (in fact the size of P_t is equal to the size of P_1), half of the Pareto front will be deleted during the transfer. This process will be continued until all individuals of a specific Pareto front cannot be completely provided in the parent population of P_t . Therefore, for choosing the exact number of individuals of that particular front for filling remained space of the population P_t , an associate population with reference points to keep a constant number of individuals (POP: population size). Finally, the population update process P_t will be used to replace population P_1 in the next generation of the SPEA/R algorithm and the Pareto temporary front optimization will be defined after a specific generation can. Therefore, after each cycle of population update (P_t) in the SPEA/R algorithm, Pareto front temporarily will be transferred to the DNN including input data and output data (that is the Temporary Pareto front) to train the DNN. From here, we see a lot of Temporary Pareto front files created from the SPEA/R algorithm, which means there are many standard data sets for training DNN. Therefore, the combined training of the DNN has the full characteristics of the SPEA/R algorithm, but it has a fast calculation speed and converges faster of neural networks. The search area is larger than the SPEA/R algorithm because the DNN is trained on many standard samples. This process has been generalized according to Algorithms 1 and 2 as follows

Algorithm 1: Training DNN optimize

- 1: Input: Pareto temporary front set
- 2: Output: weight and bias values optimized
- 3: Initializing parameters
 - Population size
 - Number of generation
 - Probabilities of selection
 - Crossover and mutation
 - Fitness function
 - Mean square error
- 4: Creating initial population parent encoding weight and bias
- 5: **while** *stopping criterion not met* **do**
- 6: Calculation fitness
- 7: Evolution of population
- 8: Fitness ranking
- 9: Selection crossover
- 10: Selection of fitness and update
- 11: Record the best chromosome
- 12: **end while**

Algorithm 2: Training DNN

- 1: **Input:** N (population size)
- 2: **Output:** Pareto-optimal front set
- 3: Create a diverse set of reference directions F
 - $F := \text{Reference Create } ();$
- 4: Create an original parent population $P1$;
- 5: **while** *stop criterion not met* **do**
- 6: Apply this genetic operator $P1$ to create offspring
- 7: $Q := P1 \cup P2$;
- 8: Normalize the goals of internal members Q :
 - $Q := \text{Objective normalization}(Q);$
- 9: **for** each reference direction $i \in F$ **do**
- 10: Identify members of Q close to i :
 - $H(i) := \text{Associate } (Q, F, i);$

- 11: Perform calculations fitness values of members in $H(i)$;
- Fitness assignment ($H(i)$);
- 12: **end for**
- 13: P_i : = Choose the environment (Q, F);
- 14: Call **Algorithm 1**
- 15: **end while**

Leave-one-out cross-validation (LOOCV) method. Cross-validation is the most common and effective way to verify models in statistical machine learning. This method is intended to estimate predictive models that they have learned from training data. How it will do on data that has not been tested in the future. In other words, by this method, we can measure the generalized power (accuracy) of our trained model in practice and avoid over-fitting when the use of techniques in cross-validation. LOOCV is the most common cross-authentication method. This technique is often applied when the amount of training and testing data is not too large or too difficult to create a large training/test data set for model learning. With this method, in each iteration, a sample of the temporary data point is considered as the validation data and the remaining data is used for model training. Through each model training process, its prediction errors will be calculated on the validation data. If the initial training data contains $I_{\text{ter_max}}$ samples, this procedure repeats $I_{\text{ter_max}}$ times (it equals the number of observations in the initial training set). Then, the average of these errors is reported as the predictive error of our predictive model across the entire data set in terms of expressions by equation (2). Thus, with this validation method, in its iterations $I_{\text{ter_max}}$, all patterns have the opportunity to act as a prototype. For more information on LOOCV, refer to Ron.²¹

$$\sum_{\text{error}} = \left(\sum_{i=1}^{I_{\text{ter_max}}} |I_{\text{ter_}i}| \right) / I_{\text{ter_max}} \tag{2}$$

Vibration characteristic of the powertrain mount system

Full car model with 10 degrees of freedom (D.O.F.) is shown in Figure 5. Suspension and tires are considered spring and damping systems. Where the masses $m_{2,1}$, $m_{2,2}$, $m_{2,3}$, and $m_{2,4}$ denote the weight of four wheels (the mass does not burst). The masses m_{cg1} and m_{cab} represent the mass (unburnt mass) of the frame and engine, respectively. $Z_{3,1}$, $Z_{3,2}$, $Z_{3,3}$, and $Z_{3,4}$ are the vertical displacements of the wheel, Z_1 , Z_2 are, respectively, the vertical displacement of the frame and the transmission system. And, Roll and Altitude are vibrations that rotate around the corresponding X and Y axes. Next, the symbols Φ and Ψ represent the pitch and roll of the frame. The inertia of the transmission system on the y -axis and the x -axis are I_{yy} and I_{xx} , respectively, the inertial

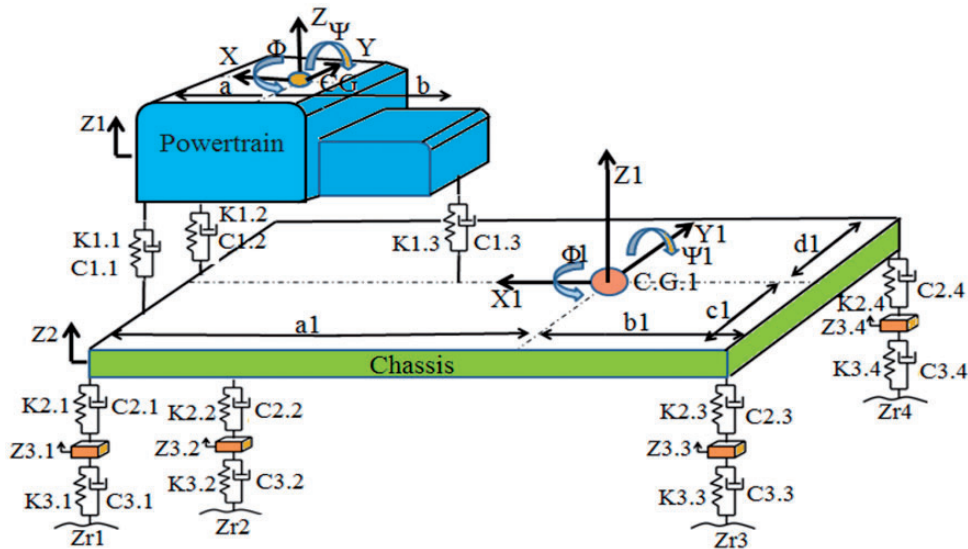


Figure 5. The principle diagram of a full-car dynamic model with a power source system mounting system.

moment for the chassis on axes $X1$ and $Y1$ are I_{yy1} and I_{xx1} , respectively. The stiffness and damping parameters of the wheels are $K3.1, K3.2, K3.3, K3.4$ and $C3.1, C3.2, C3.3, C3.4$, respectively. Similarly, the hardness and damping parameters of primary suspension are $K2.1, K2.2, K2.3, K2.4$ and $C2.1, C2.2, C2.3, C2.4$, respectively, while the hardness and damping parameters of the drive system are $K1.1, K1.2, K1.3$ and $C1.1, C1.2, C1.3$, respectively. The distance of the front and rear support from the center (CG) of the transmission system is a and b , respectively; the right and left mounting distances from the CG of the transmission system are c and d , respectively. $a1, b1, c1$, and $d1$ Distances for the chassis are indicated as follows.

By using Newton's law, the mathematical model of Figure 4 can be written as below

$$M\ddot{x}_i + K\dot{x}_i + Cx_i = Q(t) \quad (3)$$

where the symbols are shown in Table 1.

Simulation results and discussion

Numerical results

In this section, to demonstrate the superiority of the HDNN&SPEA/R method, we conducted calculations and tests on a number of benchmark functions. The results are compared to SPEA/R algorithms and Simplex NSGAI algorithms.²² For the performance metric,²³ inverted generational distance (IGD),²⁴ spacing (SP), and maximum spread (MS)²⁵ criteria are employed to measure convergence, quantity, and coverage, respectively. The mathematical formulation of IGD is as follows

$$IGD = \frac{\sqrt{\sum_{i=1}^N d_i^2}}{N} \quad (4)$$

where N is the number of true Pareto front optimal solutions, and d_i indicates the Euclidean distance between the i -th true Pareto front optimal solution and the closet obtained Pareto front optimal solutions in the reference set. It should be noted that $IGD=0$ indicates that all members of the non-dominated solutions are in the true Pareto front

$$SP = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (\bar{d} - d_i)^2} \quad (5)$$

where \bar{d} is the average of all d_i , N is the number of Pareto front optimal solutions obtained, and $d_i = \min(|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|)_j$ for all $i, j = 1, 2, 3, \dots, N$

$$MS = \sqrt{\sum_{i=1}^M \max[d(a_i, b_i)]} \quad (6)$$

Table 1. Parameters of the mathematical model.

Symbol	Parameters of the mathematical model
X_i	Vector-column of displacements and angular oscillations of masses.
M	The matrix of inertial coefficients of car parts.
C	The matrix of coefficients of stiffnesses and torsional rigidity.
K	The matrix of damping coefficients.
$Q(t)$	Column vector of the perturbing forces and moments. $q_2(t) = q_1(t + \tau)$ with τ : time interval, v : vehicle speed.

Table 2. Benchmark Functions for test multi-objective optimization.

Mathematics formulation	Mathematics formulation	Mathematics formulation
<p>UF2:</p> $f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2$ $f_2 = 1 - \sqrt{x} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $y_j = \begin{cases} x_j = \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1 \right] \sin\left(6\pi x_1 + \frac{j\pi}{n}\right) & \text{if } j \in J_1 \\ x_j = \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1 \right] \sin\left(6\pi x_1 + \frac{j\pi}{n}\right) & \text{if } j \in J_2 \end{cases}$ <p>$J_1 = \{j j \text{ is odd and } 2 \leq j \leq n\};$ $J_2 = \{j j \text{ is even and } 2 \leq j \leq n\};$</p>	<p>UF4:</p> $f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j)$ $f_2 = 1 - x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)$ $j = 2, 3, \dots, n$ $h(t) = \frac{ t }{1 + e^{2 t }}$ <p>J_1 and J_2 are the same as those of UF2</p>	<p>UF5:</p> $f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right) \sin(2N\pi x_1) + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j)$ $f_2 = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right) \sin(2N\pi x_1) + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)$ $j = 2, 3, \dots, 4$ $h(t) = 2t^2 - \cos(4\pi t) + 1$ <p>J_1 and J_2 are the same as those of UF2, $\varepsilon > 0$.</p>
<p>UF8:</p> $f_1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $f_2 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ <p>J_1, J_2, and J_3 are the same as those of UF10</p>	<p>UF10:</p> $f_1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} (4y_j - \cos(8\pi y_j + 1))$ $f_2 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} (4y_j - \cos(8\pi y_j + 1))$ $f_3 = \cos(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} (4y_j - \cos(8\pi y_j + 1))$ <p>$J_1 = \{j 3 \leq j \leq n \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j 3 \leq j \leq n \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j 3 \leq j \leq n \text{ and } j-3 \text{ is a multiplication of } 3\}$</p>	

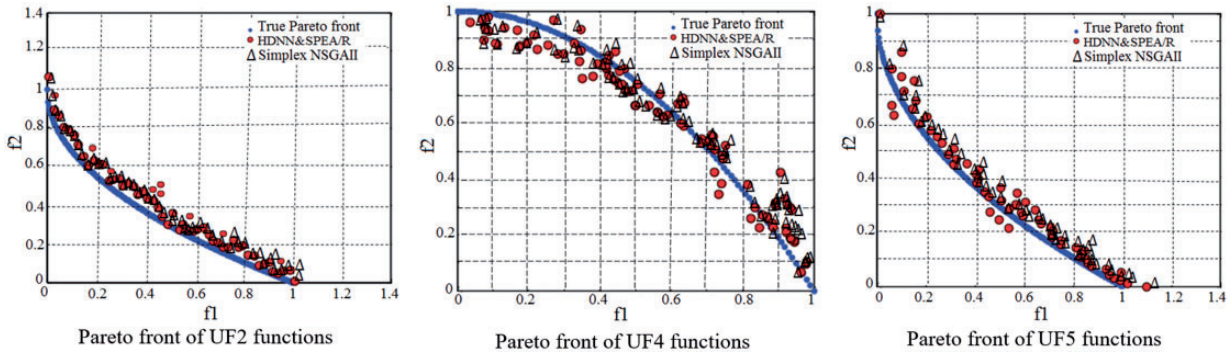


Figure 6. Pareto front set of bi-objective benchmark functions.

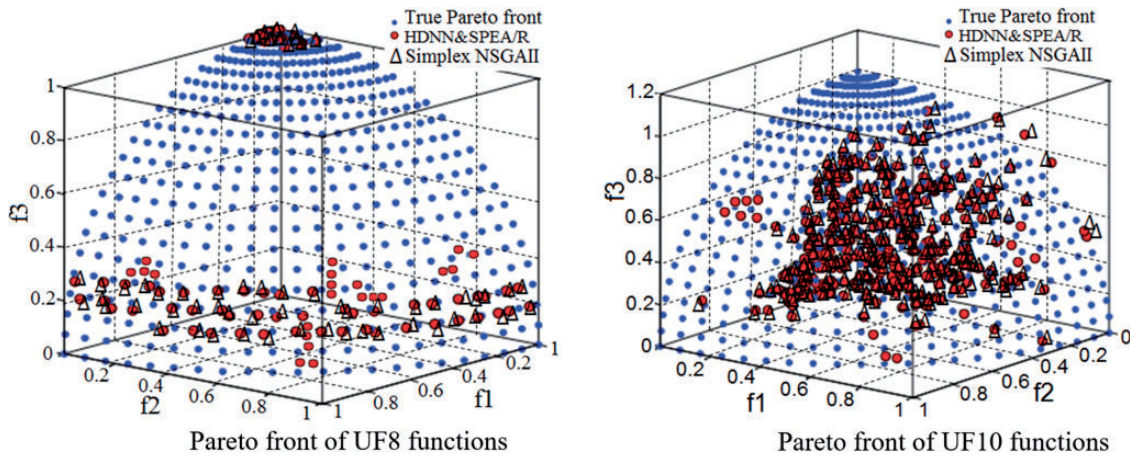


Figure 7. Pareto front set of tri-objective benchmark functions HDNN&SPEA/R.

where d is a function to calculate the Euclidean distance, b_i is the maximum value in the i -th objective, b_i is the minimum in the i -th objective, and M is the number of objectives.

In addition to utilizing the performance metrics, the best set of Pareto front optimal solution obtained by HDNN&SPEA/R on both the parameter space as well as the search space is shown in Figures 6 and 7. These figures show the performance of HDNN&SPEA/R in comparison with the true Pareto front, and at the same time it is compared with the results of Simplex NSGAI algorithms. For a comparative evaluation, all the algorithms are run 20 times on the test problems, and the statistical results of the 20 runs with algorithm parameters are provided in Table 3. Statistical results of the algorithm for IGD, SP, and MS are provided in Tables 4 to 6, respectively of benchmark functions. In Table 4, the IGD parameter is a performance indicator that shows the accuracy and convergence of the algorithm. The IGD parameter shows that the proposed hybrid algorithm (HDNN&SPEA/R) can provide better results than the Simplex NSGAI algorithm, but it is far more than the SPEA/R algorithm.

Similarly, parameters SP and MS in Tables 5 and 6 also show that HDNN&SPEA/R algorithm is the most dominant. Pareto optimal solution of HDNN&SPEA/R on each benchmark is also compared with the standard Pareto front set and the result of Simplex NSGAI algorithm as described in Figures 6 and 7.

Simulation results test benchmark functions

The resulting Pareto front is shown in Figures 6 and 7.

In this regard, Figures 6 and 7 show the comparison between the predicted results of HDNN&SPEA/R algorithms with the result of Simplex NSGAI algorithm being similar, but in terms of coverage, HDNN&SPEA/R algorithm is somewhat more outstanding. The results of both algorithms closely follow the

Table 3. Model parameters of SPEA/R algorithm and DNN

Maximum number of iterations	Iter_max = 1000	pMutation	1-pCrossover
Population size	100	nMutation	nPop-nCrossover
Archive size	50	mutation_params.h	0.2
KNN parameter	round(sqrt(nPop + nArchive))	crossover_params.gamma	0.1
pCrossover	0.7	crossover_params.VarMin	VarMin
nCrossover	round(pCrossover × nPop/2) × 2	mutation_params.VarMin	VarMin
mutation_params.VarMax	VarMax	crossover_params.VarMax	VarMax
DNN parameters	Nn = (100 200 200 200 200 100)		

DNN: deep neural network, KNN: K-Nearest Neighbor.

Table 4. Results for IGD.

IGD	UF2					UF4				
	Average	Median	Std. dev	Worst	Best	Average	Median	Std. dev	Worst	Best
Simplex NSGAI1	0.01359	0.01225	0.00289	0.01845	0.01060	0.02434	0.02415	0.00138	0.02760	0.02295
SPEA/R	0.12244	0.1242	0.01243	0.14485	0.10454	0.06823	0.06835	0.00254	0.07078	0.06424
HDNN&SPEA/R	0.01351	0.01544	0.00245	0.01456	0.01274	0.02645	0.02867	0.00145	0.02734	0.02267
	UF5					UF8				
IGD	Average	Median	Std. dev	Worst	Best	Average	Median	Std. dev	Worst	Best
Simplex NSGAI1	0.45289	0.45486	0.08449	0.53701	0.25741	0.19992	0.20740	0.06217	0.33433	0.12553
SPEA/R	1.26755	1.33741	0.13839	1.46735	0.12145	0.56681	0.53667	0.28667	0.69647	0.28530
HDNN&SPEA/R	0.47843	0.45450	0.08445	0.53561	0.22371	0.19639	0.27870	0.06765	0.33467	0.17565
	UF10									
IGD	Average	Median	Std. dev	Worst	Best					
Simplex NSGAI1	1.70149	1.54063	0.55163	3.02835	1.12806					
SPEA/R	1.63529	1.59123	0.29349	2.16232	1.22048					
HDNN&SPEA/R	1.70344	1.54333	0.55135	3.03833	1.13821					

IGD: inverted generational distance.

results of benchmarking as shown in Figures 6 and 7. Therefore, it can be said that the proposed hybrid technique of HDNN&SPEA/R in this article has been successfully implemented.

The numerical results demonstrate that HDNN&SPEA/R is proposed to have better performance for benchmarking functions with two objectives related to convergence and coverage compared to SPEA/R algorithms. For some tri-objective benchmark functions, the proposed algorithm shows high convergence; it has high coverage. Thus, we could say the key advantages of the proposed HDNN&SPEA/R algorithm compared to SPEA/R and Simplex NSGAI1 algorithm are high convergence and coverage characteristics.

Simulation results test optimization of the Powertrain Mount System Stiffness Parameter

When the vehicle moves there are many factors that cause the vibration, the factors can be told: the internal force in the car; external forces appear in the process of using acceleration, braking, and revolving; exterior conditions such as wind and storm; and boring face street. Among the other factors is the bumpy side of the road which causes oscillation of the vehicle. To simulate the most general calculation, we use the road surface profile as a random function as in Figure 8 and simulated parameters as shown in Table 7.

Through Matlab software and SPEA/R and HDNN&SPEA/R algorithms for simulation calculation to find Pareto front optimal set simultaneously of six functions of acceleration and displacement according to the

Table 5. Results for SP.

SP	UF2					UF4				
	Average	Median	Std. dev	Worst	Best	Average	Median	Std. dev	Worst	Best
Simplex NSGAI	0.02770	0.02654	0.01289	0.06199	0.01640	0.02543	0.02386	0.00475	0.03384	0.02055
SPEA/R	0.00866	0.00879	0.00096	0.01042	0.00797	0.00780	0.00758	0.00066	0.00876	0.00617
HDNN&SPEA/R	0.02743	0.02634	0.01269	0.06449	0.01754	0.02865	0.02865	0.00478	0.03743	0.02054
SP	UF5					UF8				
	Average	Median	Std. dev	Worst	Best	Average	Median	Std. dev	Worst	Best
Simplex NSGAI	0.12049	0.12758	0.02505	0.15826	0.07457	0.23364	0.23699	0.03859	0.27492	0.16389
SPEA/R	0.00278	0.00007	0.00553	0.01665	0.00001	0.02692	0.02737	0.00867	0.04453	0.01731
HDNN&SPEA/R	0.12645	0.12433	0.03455	0.16854	0.7454	0.24367	0.23849	0.03845	0.28652	0.16229
SP	UF10									
	Average	Median	Std. dev	Worst	Best					
Simplex NSGAI	1.06788	0.89524	0.41661	1.81766	0.67219					
SPEA/R	0.01984	0.02064	0.00378	0.02645	0.01538					
HDNN&SPEA/R	1.06754	0.89523	0.41943	1.81745	0.67354					

SP: spacing.

Table 6. Results for MS.

MS	UF2					UF4				
	Average	Median	Std. dev	Worst	Best	Average	Median	Std. dev	Worst	Best
Simplex NSGAI	0.89222	0.88328	0.05448	0.82385	0.93883	0.95759	0.96323	0.01762	0.92252	0.97665
SPEA/R	0.87501	0.87337	0.00580	0.85436	0.87884	0.88564	0.88431	0.01866	0.84324	0.91394
HDNN&SPEA/R	0.89542	0.88678	0.05765	0.82565	0.93433	0.95656	0.96873	0.01542	0.92642	0.95676
MS	UF5					UF8				
	Average	Median	Std. dev	Worst	Best	Average	Median	Std. dev	Worst	Best
Simplex NSGAI	0.83746	0.83556	0.06986	0.69914	0.95996	0.35198	0.34883	0.44928	0.27879	0.73412
SPEA/R	0.29341	0.29565	0.3560	0.23544	0.34550	0.55310	0.54401	0.16736	0.26523	0.71676
HDNN&SPEA/R	0.82456	0.83646	0.06756	0.64514	0.95656	0.34648	0.36343	0.47768	0.27859	0.77772
MS	UF10									
	Average	Median	Std. dev	Worst	Best					
Simplex NSGAI	0.27643	0.14774	0.38569	0.03183	0.93133					
SPEA/R	0.13435	0.10433	0.06253	0.06679	0.25414					
HDNN&SPEA/R	0.24632	0.14324	0.38439	0.03433	0.96433					

MS: maximum spread.

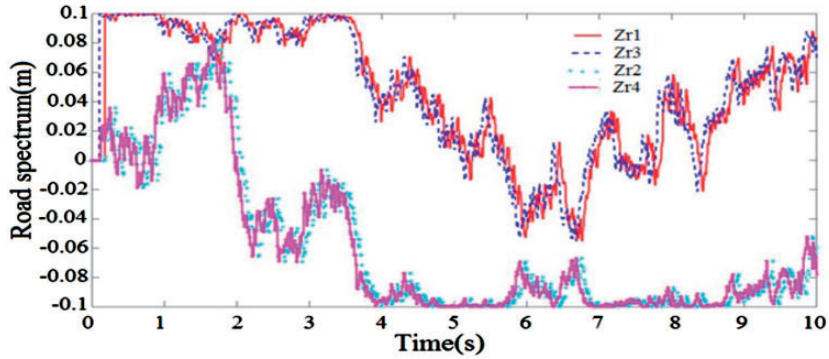


Figure 8. Road surface profiles.

Table 7. Model parameters of the powertrain mount system.

Model parameters of the powertrain mount system			
Numbers	Parameters	Values	Units
1	$m_{2.1}, m_{2.2}, m_{2.3},$ and $m_{2.4}$	60	Kg
2	m_{cg1} and m_{cab}	1000, 1200	Kg
3	$K2.1, K2.2, K2.3, K2.4$	37,000	N/m
4	$C2.1, C2.2, C2.3, C2.4$	700	N-s/m
5	$K3.1, K3.2, K3.3, K3.4$	55,000	N/m
6	$C3.1, C3.2, C3.3, C3.4$	4000	N-s/m
7	$K1.1, K1.2, K1.3$	67,000	N-s/m
8	$C1.1, C1.2, C1.3$	6000	N-s/m
9	$a, b, c = d$	0.187, 0.623, 0.3	M
10	$d1 = b1, c1 = d1$	1.5, 1	M
11	l_{xx}, l_{yy}	320, 80	Kg-m ²
12	l_{xx1}, l_{yy1}	4000, 950	Kg-m ²

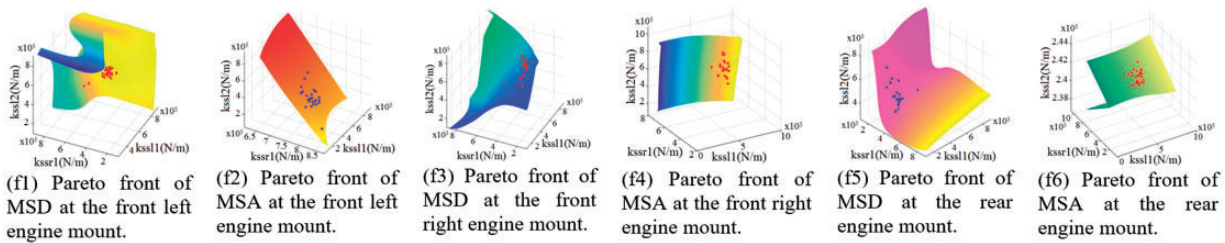


Figure 9. Global Pareto front set of six-objective optimization functions with SPEA/R.

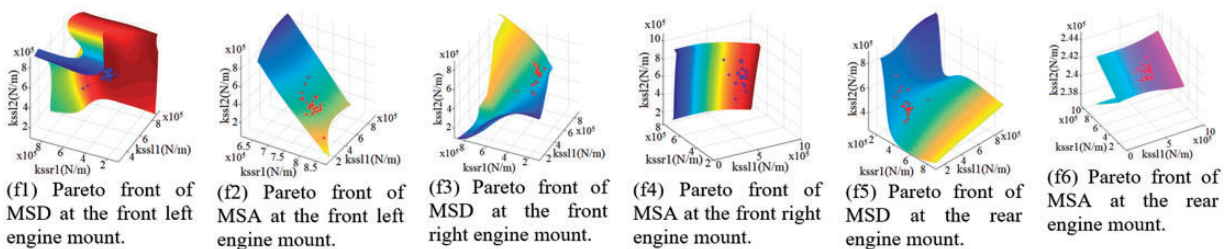


Figure 10. Global Pareto front set of six-optimization objective with HDNN&SPEA/R.

Table 8. Results for IGD of Powertrain mount system.

IGD	Powertrain mount system				
	Average	Median	Std. dev	Worst	Best
SPEA/R	2.54558	2.68427	0.37349	3.86259	1.92940
HDNN&SPEA/R	2.91574	2.84373	0.381459	4.05993	2.14391

IGD: inverted generational distance.

Table 9. Results for SP of Powertrain mount system.

SP	Powertrain mount system				
	Average	Median	Std. dev	Worst	Best
SPEA/R	0.55955	0.42976	0.50478	0.92846	0.64569
HDNN&SPEA/R	0.86050	0.69569	0.61845	1.51648	0.87655

SP: spacing.

Table 10. Results for MS of Powertrain mount system.

MS	Powertrain mount system				
	Average	Median	Std. dev	Worst	Best
SPEA/R	0.32338	0.36429	0.52268	0.13476	0.67448
HDNN&SPEA/R	0.34739	0.48362	0.68488	0.15453	0.87498

MS: maximum spread.

Table 11. Results for time analysis compare of Powertrain mount system.

Results for time analysis compare of Powertrain mount system (h)										
Test times	1	2	3	4	5	6	7	8	9	10
SPEA/R	30.12	30.04	29.80	30.06	29.70	31.01	30.06	30.15	29.96	29.89
HDNN&SPEA/R	1.58	1.45	1.52	1.59	1.40	1.49	1.60	1.47	1.49	1.48
Test times	11	12	13	14	15	16	17	18	19	20
SPEA/R	30.50	30.02	30.08	29.95	30.02	30.55	30.45	29.08	29.99	30.30
HDNN&SPEA/R	1.51	1.53	1.54	1.52	1.51	1.51	1.55	1.54	1.53	1.52
Total squared average	SPEA/R (30.0865 h): HDNN&SPEA/R(1.5165 h)									

stiffnesses of the front left, front right, and rear engine mount ($K_{1.1}$, $K_{1.2}$, and $K_{1.3}$) value as shown in Figures 9 and 10 (Figures 9 and 10 shows the results in the form of 4D via the isosurfaces function in Matlab).

In order for a more intuitive look, the analytical results are presented in statistical form in the Tables 8 to 11.

It is known that the result of multi-objective optimization will be a set of unaffected optimization points, called Pareto sets. These points provide a series of parameters for the designer to select the optimal point depending on his design conditions. There are always conflicting objective functions in the vehicle design that improving a function can adversely affect other functions. In this article, multi-objective optimization for all six target functions is done simultaneously. With the two results obtained from the two algorithms shown in Figures 9 and 10, we see Pareto front optimal set of two algorithms found relatively similar. But HDNN&SPEA/R algorithm has very fast computation time compared to SPEA/R algorithm. Therefore, through tests comparing two algorithms, HDNN&SPEA/R algorithms have proven superiority over SPEA/R algorithm because the time of calculation is extremely small, and the accuracy is completely reliable. That is, the optimal calculation time of HDNN&SPEA/R is much less than the SPEA/R algorithm. The results of the proposed HDNN&SPEA/R were in almost all cases better than SPEA/R.

Conclusion

This article has published a new combination method between SPEA/R, DNN, and GA. So ANNs with many hidden layers are trained with intelligent deep learning algorithms; it has created a deep learning network to realize the optimal problem simultaneously of many objects in the technology. This method is most effective and highly practical because: firstly, it offers a way of deep learning of the network with a much smaller number of standard samples than previous deep-learning networking methods announced. For test cases optimization of the Powertrain Mount System Stiffness Parameter, we only need a standard set of 15 samples to train DNNs. It is this combination of training that the actual number of samples generated for the training process is $15 \times (\text{Iter_max}) = 15,000$ samples. While training with the old method is extremely difficult to collect a large number of samples. Secondly, for test cases optimization of the Powertrain Mount System Stiffness Parameter, the time for multi-objective optimal analysis of HDNN&SPEA/R hybrid methods is only 1.5 h, while using SPEA/R algorithm takes 30 h.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (51875096, 51275082).

ORCID iD

Dinh-Nam Dao  <https://orcid.org/0000-0001-9810-2096>

References

1. De Tommasi L, Beelen TGJ, Sevat MF, et al. Multi-objective optimization of RF circuit blocks via surrogate models and NBI and SPEA2 methods. CASA-report, Technische Universiteit Eindhoven, Eindhoven, 2011, vol. 1132.
2. Sofianopoulos S and Tambouratzis G. Studying the SPEA2 algorithm for optimising a pattern-recognition based machine translation system. In: *Proceedings of the 2011 IEEE symposium on computational intelligence in multicriteria decision-making (MCDM 2011)*, Paris, France, 11–15 April 2011, pp. 97–104.
3. Zhao F, Lei W, Ma W, et al. An improved SPEA2 algorithm with adaptive selection of evolutionary operators scheme for multiobjective optimization problems. *Math Probl Eng* 2016; 2016: 20.
4. Ben Hamidabrini Salah I, Msahli SF, et al. Strength Pareto evolutionary algorithm 2 for environmental/economic power dispatch. In: *ICMIC 2015: 7th International conference on modelling, identification, and control*, Tunisia, 18–20 December 2015. pp(65–76). Elsevier Ltd.
5. Jiang S and Yang S. A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Trans Evol Comput* 2017; 21: 329–346.
6. Mahmoud Shakouri Hassanabadi & Saeed Banihashemi Namini (2012): Developing an empirical predictive energy-rating model for windows by using Artificial Neural Network, *International Journal of Green Energy*. DOI:10.1080/15435075.2012.738451.
7. Hansen K, et al. Machine learning predictions of molecular properties: accurate many-body potentials and nonlocality in chemical space. *J Phys Chem Lett* 2015; 6: 2326–2331.
8. Montavon G, et al. Learning invariant representations of molecules for atomization energy prediction. In: Pereira F, Burges CJC, Bottou L, et al. (eds) *Advances in neural information processing systems*. Curran Associates, Inc., NY, USA, 2012, pp. 440–448.
9. Chan KY, Dillon T, Chang E, et al. Prediction of short-term traffic variables using intelligent swarm-based neural networks. *IEEE Trans Contr Syst Technol* 2013; 21: 263–274.
10. Leelavathi MV and KJ SD. An architecture of deep learning method to predict traffic flow in big data. *IJRET* 2016; 5: 461–468.
11. Huang W, Song G, Hong H, et al. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Trans Intell Transport Syst* 2014; 15: 2191–2201.
12. Wang J, Wang J, Fang W, et al. Financial time series prediction using elman recurrent random neural networks. *Comput Intell Neurosci* 2016. <http://dx.doi.org/10.1155/2016/4742515>
13. Sreekanth J and Datta B. Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network-based surrogate models. *J Hydrol* 2010; 393: 245–256.

14. Gene L and Steven C. Multi-objective evolutionary neural network to predict graduation success at the United States Military Academy. In: Cihan H Dagli (ed) *Complex adaptive systems conference with theme: cyber physical systems and deep learning*, CAS, Chicago, Illinois, USA, 5–7 November 2018.
15. Shen Y, Wang X and Chen J. Wind power forecasting using multi-objective evolutionary algorithms for wavelet neural network-optimized prediction intervals. *Appl Sci* 2018; 8: 185.
16. Smith C, Doherty J and Jin Y. Multi-objective evolutionary recurrent neural network ensemble for prediction of computational fluid dynamic simulations. In: J Schmidhuber (ed) *IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, 6–11 July 2014. DOI: 10.1109/CEC.2014.6900552.
17. Kakaee A-H, Rahnama P and Paykani A. Combining artificial neural network and multi-objective optimization to reduce a heavy-duty diesel engine emissions and fuel consumption. *J Cent South Univ* 2015; 22: 4235.
18. Vieira A and Tome RS. A multi-objective evolutionary algorithm using neural networks to approximate fitness. *Int J Comput Syst Sign* 2005; 6: 18–36.
19. Tansel IN, Gulmez S, Demetgul M, et al. Taguchi method–GONNS integration: complete procedure covering from experimental design to complex optimization. *Expert Syst Appl* 2010; 38: 4780–4789.
20. Filho ACP and Filho RM. Hybrid training approach for artificial neural networks using genetic algorithms for the rate of reaction estimation: application to industrial methanol oxidation to formaldehyde on the silver catalyst. *Chem Eng Sci* 2010; 157: 501–508.
21. Ron K. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: J Schmidhuber (ed) *Proceedings of the fourteenth international joint conference on artificial intelligence*, 1995, pp.1137–1143.
22. Pourtakdoust SH and Zandavi SM. A hybrid simplex non-dominated sorting genetic algorithm for multi-objective optimization. *Int J Swarm Intel Evol Comput* 2016; 5: 138.
23. Pradhan PM and Panda G. Solving multi-objective problems using cat swarm optimization. *Expert Syst Appl* 2012; 39: 2956–2964.
24. Schott JR. Fault-tolerant design using single and multi-criteria genetic algorithm optimization, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology.1995. Includes bibliographical references pp. 199–200, <http://hdl.handle.net/1721.1/11582>.
25. Zitzler E and Thiele L. Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 1999; 3: 257–271.