

RESEARCH

Open Access

Research on music creation and audition based on wireless environment



Lingzi Wang and Mingxia Pan*

* Correspondence: xingbaifu19811@163.com
Academy of Arts, West Anhui
University, Lu'an 237009, China

Abstract

Traditional computer music creation is carried out through computer simulation of fixed conversion of syllables. However, there are phenomena of syllables and pitch distortions in music creation that changes rapidly in rhythm. In response to this problem, a method of music creation and sound test based on genetic algorithm and wireless network is proposed. First, the genetic algorithm is introduced to construct a general mathematical module for music creation, and genetic operators for encoding string length and group size are assigned to establish a similarity measure for computer music creation. Then, the mean cluster analysis of music creation realizes the creation of computer music. Finally, based on the wireless network technology, the sound test effect analysis of music creation was carried out. Experimental results show that the proposed method of music creation and sound test is superior to the traditional method, the accuracy of syllables is increased by 25.97%, and the accuracy of tones is increased by 34.12%. This technology can realize high-pitched music creation and audition research.

Keywords: Music creation, Genetic algorithm, Wireless network, Mean clustering

1 Introduction

In traditional computer music creation, music is constructed by computer simulation of fixed conversion of syllables and by specific databases and calling functions. However, when the rhythm changes greatly, the frequency will increase, which will cause the tone function to oversaturate and distort the syllable and pitch [1]. In response to this problem, a method of music creation and sound test based on genetic algorithm and wireless network is proposed. First, the article establishes a general mathematical module for music creation. Genetic algorithm is introduced to establish the relevant constraint equations for computer music creation, as well as the chord length, population size, genetic evolution termination algebra, crossover probability, and mutation probability. Judging the relative relationship based on the mean clustering calculation of computer music creation, the best solution of Pareto is used to realize computer music creation, thus, completed the research of computer music creation based on genetic algorithm [2].

In addition, in order to ensure the effectiveness of the proposed computer music creation method, this paper simulates the music type test environment. Music audition

work was carried out by a method based on efficient data transmission of wireless sensor network. Two different computer music creation methods are used to simulate syllable pitch accuracy [3]. The experimental results show that the proposed computer music creation and sound test methods are effective.

2 Methods

2.1 Building of mathematical module of computer music creation

In order to solve the problem of computer music creation, a general mathematical module of computer music creation is built. First, the decision variable is determined. Generally, in the computer music creation, the decision variable is regarded as a point of E^n in Euclidean space, which is expressed as

$$x = (x_1, x_2, \dots, x_n) \in E^n \quad (1)$$

Second, the objective function of computer music creation is determined. Assume there are P objective functions of the decision variable, which is expressed as

$$f(x) = E^n [f_1(x), f_2(x), \dots, f_p(x)]^T \quad (2)$$

Finally, the constraints are defined. From a mathematical point of view, there are two constraints: inequality constraints and equality constraints. The constraints can be defined as m inequality constraints and K equality constraints. The constraints of computer music creation are given by

$$\begin{cases} g_i(x) \leq 0, i = 1, 2, \dots, m \\ h_j(x) = 0, j = 1, 2, \dots, k \end{cases} \quad (3)$$

If all the multi-objective functions of computer music creation are to obtain minimum, then the multi-objective problem can be described as the following mathematical modules, where l objective functions are solved for the maximum while $s + l - P$ objective functions are for minimum. In order to briefly express the mathematical module of multi-objective problem of computer music creation [3], a general mathematical module is given, which can be simplified as

$$\min_{x \in R} f(x) = \{X \in E^n | g_i(x) = 0, h_j(x) = 0\} \quad (4)$$

In most cases, the objectives of the multi-objective problem of computer music creation are conflicting. The improvement of one computer music sub-objective may cause the performance of other sub-objectives to be reduced [4]. It is impossible to achieve multiple objectives at the same time. Therefore, when solving the multi-objective problem, we can only coordinate and deal with each sub-objective of computer music creation so that all the sub-objective functions are as optimal as possible. There are essential differences between multi-objective problem and single-objective problem. In order to solve the optimal solution of multi-objective problem correctly, the basic concepts of multi-objective of computer music creation are discussed systematically. Assume n -dimensional real Euclidean space, given by

$$x = [x_1, x_2, \dots, x_n]^T, y = [y_1, y_2, \dots, y_n]^T \in E^n \tag{5}$$

$X \subseteq R^m$ is the constraint set of multi-objective model, $f(x) \in R^p$ is the objective function, $x_1 = X, x_2 = X, f(x) \in R'$ is the objective function, and $x \in X, x'$ is superior to all the other points in X . Then, the x is the optimal solution of the multi-objective minimization model. The optimal solution x to the multi-objective problem of computer music creation is to make every sub-objective of the objective function $f(x)$ at the same time achieve optimal solution, as shown in Fig. 1 [5].

If $\chi \in x$ and there is no x superior to χ , then, x is the Pareto optimal solution of multi-objective minimization model of computer music creation, which is called noninferior solution, as shown in Fig. 2 [6].

It is a set of Pareto optimal solutions to create the multi-objective of computer music creation. In most cases, the optimal solution of a single objective problem does not exist in the multi-objective problem, and only the Pareto optimal solution exists. The Pareto optimal solution of multi-objective problem is only an acceptable and not bad solution, and most of the multi-objective problem usually has multiple Pareto optimal solutions [7]. The noninferior set and the optimal front end of computer music creation are shown in Fig. 3.

In practical application, according to the degree of understanding of the problem and the personal preference of the decision-maker, one or more solutions are selected from the set of Pareto optimal solutions for the multi-objective of computer music creation as the optimal solution of the multi-objective problem. So, the first step to solve the multi-objective problem is to find all the Pareto optimal solutions [8].

2.2 Establishment of music audition model based on wireless network

Based on the above Pareto best solution for computer music creation model design, this section will carry out wireless network-based music audition

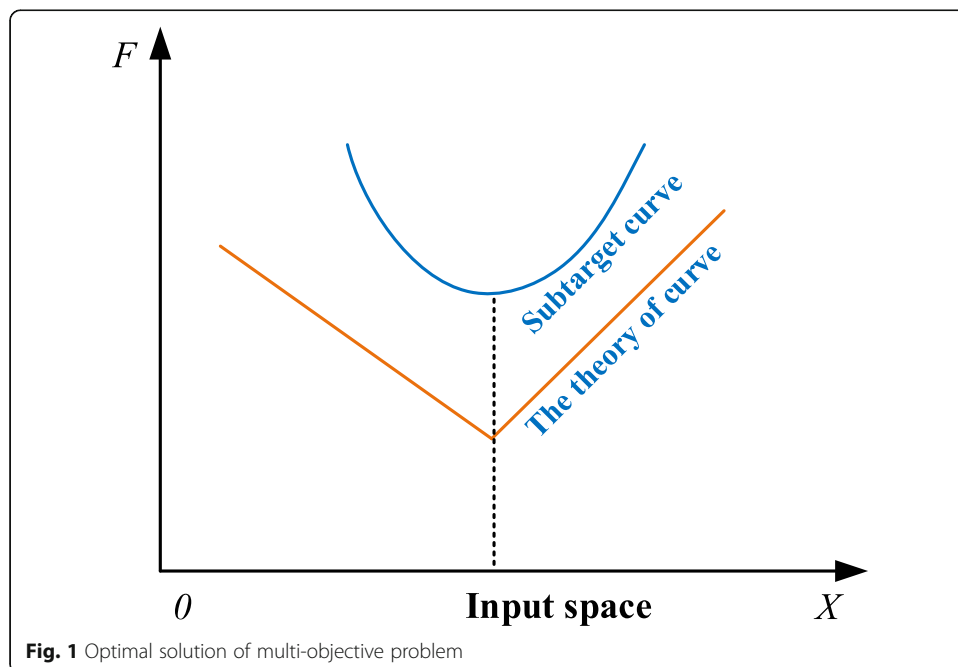
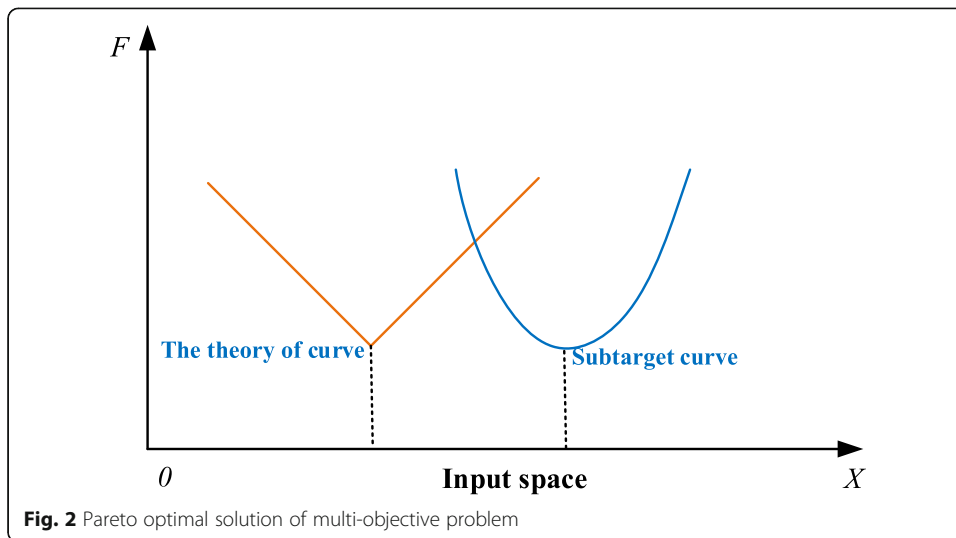
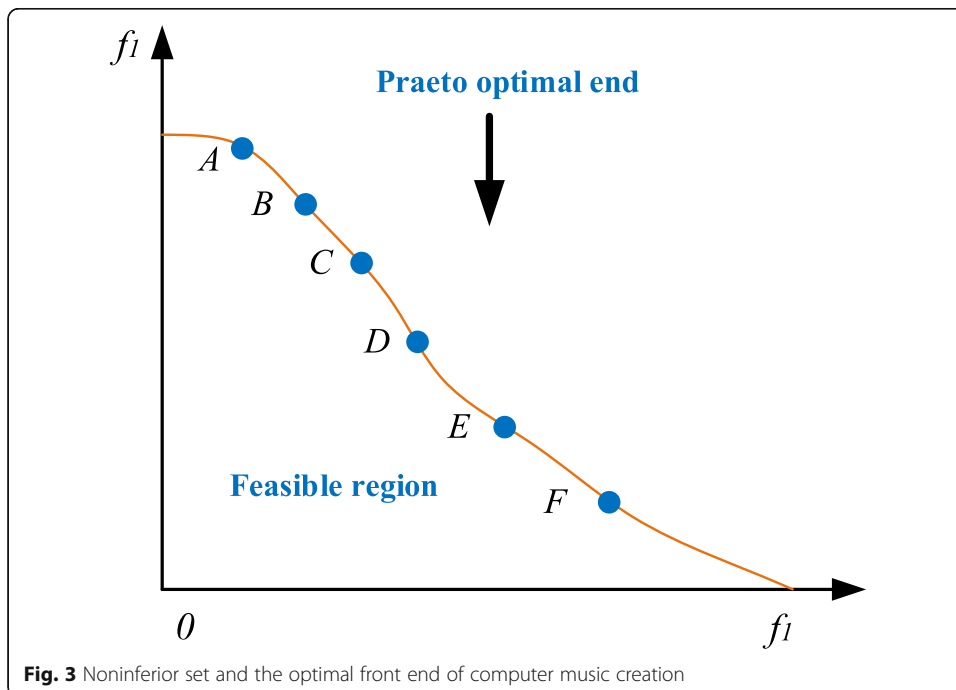


Fig. 1 Optimal solution of multi-objective problem



model design. In order to ensure the effectiveness of the proposed computer music creation method, this paper simulates a music type test environment. Music audition work was carried out by a method based on efficient data transmission of wireless sensor network. Two different computer music creation methods are used to simulate syllable pitch accuracy. The experimental results show that the proposed computer music creation and sound test methods are effective (Fig. 4).

Since the purpose of the sensor network deployment is generally to collect music audition data back, this determines a typical feature of the network transmission is the unidirectionality of a large amount of data, that is, a large amount



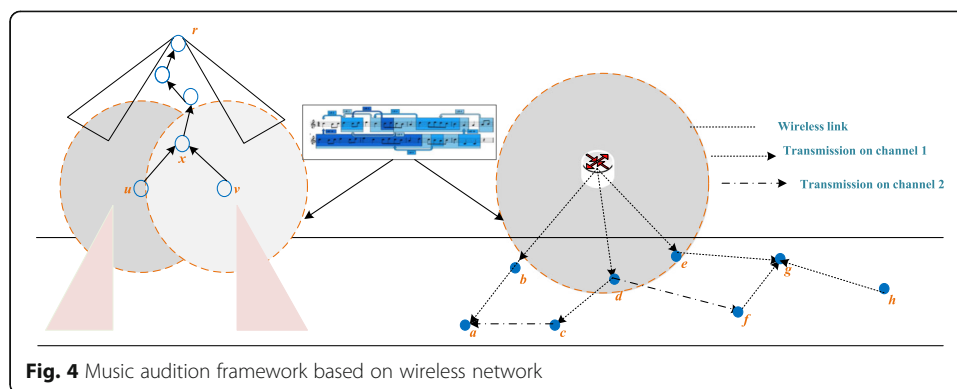


Fig. 4 Music audition framework based on wireless network

of data is aggregated from the sensor-equipped terminal network node transmission, and the energy of these terminal nodes is often the most limited.

Error control in a broad sense should include error control coding and feedback, retransmission, and other protocols. In fact, these methods cannot avoid the additional calculation and energy requirements for the data sender. Taking the typical ARQ protocol as an example, due to the channel quality or the receiver's problem causing the received data to be wrong, the sender is required to re-transmit the original data. This way invisibly wastes the sender's energy resources, and in order to avoid the channel repeated retransmissions under bad conditions require the sender to perform complex channel coding to improve the transmission success rate, which also reduces the sender's survival time to a certain extent. Therefore, the energy consumption problem caused by the above error control is more significant for this type of network. The method of solving the energy consumption caused by retransmission is generally considered to use error control coding with strong error correction capability, but as mentioned above, this also puts additional calculation requirements and transmission energy consumption on the sending end. Therefore, designing a method that does not rely on coding while reducing retransmission is necessary to improve the overall energy efficiency of the network. This article is based on this consideration, from the perspective of reducing the energy consumption of the front-end information collection node calculation and transmission as much as possible, through the wireless network transmission technology for music audition research.

3 Genetic algorithms

3.1 Genetic algorithm theory

It is a very effective way to solve the multi-objective problem of computer music creation by using genetic algorithm. The genetic algorithm is stable and robust and not easily influenced by the change of the external environment during the operation. It is especially suitable for the creation of some large and complex computer music. For some nonlinear and multi-objective optimization problems and the NP complete problem in combinatorial optimization, the traditional optimization method is difficult to solve the problem, or the quality of the solution is not ideal, but the genetic algorithm can obtain better results. Because genetic algorithm is the evolutionary operation of the whole computer music creation group [9], it is based on the individual set. From the

definition of optimal solution of multi-objective optimization, the Pareto optimal solution of multi-objective optimization problem is generally a set, so it is an effective means to solve the Pareto optimal solution of multi-objective optimization problem by genetic algorithm.

The sub-objective functions $f_i(x)(i = 1, 2, \dots, p)$ are weighted by the weights $w_i(x)(i = 1, 2, \dots, p)$. The value of w_i represents the importance of the corresponding sub-objective $f_i(x)$ in multi-objective optimization problem. The linear weighted sum of each sub-objective function is expressed as

$$u(f(x)) = w_1f_1(x) + w_2f_2(x) + \dots + w_pf_p(x) = \sum_{i=1}^p w_if_i(x) \quad (6)$$

where $w_i \geq 0(i = 1, 2, \dots, p)$ and $\sum_{i=0}^p w_i = 1$.

If the linear weighted sum of computer music creation is taken as the evaluation function of the multi-objective optimization problem, the multi-objective optimization problem can be transformed into a single-objective optimization problem. The main advantages of the algorithm are simplicity, feasibility, and low time complexity and help decision-makers to find their preferred solutions. The drawback is that when we do not know enough about the problem, it is difficult to determine the appropriate weight coefficient for each objective function. In addition, the algorithm is very sensitive to the shape of the Pareto optimal front end and cannot process the concave part of the front end.

All the individuals in the computer music creation group are equally divided into sub-populations according to the number of sub-objective functions. A sub-objective function is assigned to each sub-group [10]. Each sub-objective function independently implements selection operation in its corresponding sub-population. A number of individuals with higher fitness are selected to form a new group of computer music creation, and then all these newly generated sub-populations are merged into a complete population. In this complete population, crossover operation and mutation operation are applied to generate the next generation of complete population. In this way, the Pareto optimal solution of multi-objective optimization problems can be finally obtained through the continuous “split-parallel selection-merge” process [11].

The individuals in computer music creation group are sorted from superior to inferior. These individuals are then selected according to the probability of being large to small and satisfying the linear relationship, making the former individuals more chance to inherit the next generation. After several generations, the Pareto optimal solution of the multi-objective optimization problem of computer music creation can be finally obtained. In order to evaluate the performance of each individual in computer music creation population, a ranking method Rank is adopted. In a generation, the rank of an individual is consistent with the number of the dominated individuals in the current group [12]. In the t -th generation, all uncontrolled computer music creation individuals are ranked. If the individual x_i is dominated by p_{iD} then

$$\text{Rank}(x_i, t) = 1 + p_{iD}^t \quad (7)$$

Because this ranking method only measures the superior order between individuals, it is easy to generate a lot of similar Pareto optimal solutions, but it is difficult to

generate a more widely distributed Pareto optimal solution. In order to overcome this situation, Fonseca and Fleming adopted the sharing function and niche technology to solve the problem of group diversification. It also provides high efficiency and relatively easy implementation [13].

3.2 Selection of genetic operator of computer music creation

3.2.1 Selection operator

In the process of genetic and natural evolution of organisms, there will be more chances for the species with higher adaptation to the environment to inherit to the next generation. Like the evolution process, genetic algorithm selects the operator to select the survival of the fittest. The individuals with higher fitness are more likely to be inherited to the next generation of population, and the lower fitness individuals are less likely to be inherited to the next generation [14]. The selection operation in genetic algorithm is used to determine how to select a number of genetic operations from a parent group to the next generation group according to a certain rule.

The selection operation is based on the evaluation of the fitness of the individual. The main purpose of selection operation is to avoid gene deletion, improve global convergence, and computational efficiency. Generally speaking, the selection operator can be divided into three types: the selection based on the fitness ratio, the selection based on the adaptive value sorting, and the selection based on the individual competition [15].

Selection operator based on fitness ratio is the most commonly used selection operator in genetic algorithm. It includes roulette selection, deterministic sampling selection, no return random selection, and no return residue selection. The commonness of the selection method based on the fitness ratio is that the probability of individuals being selected and inherited to the next generation group is directly proportional to the fitness of the individual. If the adaptation of an individual in the early stage of the algorithm is much higher than that of other individuals, the individual is overly selected which leads to too fast convergence and the premature problem of genetic algorithms [16]. We should avoid this happening in multi-objective optimization genetic algorithm.

The main idea of selection operator based on fitness ranking is to rank all individuals according to their fitness. Based on this sort, the probability of each individual being selected is allocated. The advantage is that there is no special requirement for individual fitness to be positive or negative and the numerical difference between individual fitness values. The selection pressure can be increased when the diversity of the group is poor, and the selection pressure can be reduced when the diversity of the group is rich so that the premature problem of the algorithm can be avoided to some extent. The drawback is that a probability allocation table must be pre designed, and there is no certain rule in the design process. Besides, sorting selection still has a large selection error [17]. Selection operator based on individual competition mainly refers to tournament selection. The basic idea is that the individual with the highest fitness is inherited to the next generation. This method has only the comparison between individual fitness values, without the arithmetic operation between individual fitness, so there is no

special requirement for the positive or negative value of the individual fitness. For the multi-objective optimization problem, a large number of individuals should be evaluated and compared with each other for each operation, which makes the search efficiency of the algorithm lower [18].

3.2.2 Crossover operator

In the process of natural evolution, two homologous chromosomes are reorganized by mating to form new chromosomes, thus producing new individuals or species. Mating recombination is a major step in the process of biological inheritance and evolution. Crossover operator is also used to generate new individuals in genetic algorithm. Crossover operation refers to the exchange of some genes between two paired chromosomes in some way, thus forming two new individuals. Crossover operation is an important feature of genetic algorithm which is different from other evolutionary algorithms. In genetic algorithm, we must first pair the individuals in the group before crossover operation. The common pairing strategy is random pairing. The M individuals in the group are composed of paired pairs of individuals in a random way. Crossover operation is carried out among two individuals in the paired population.

The design and implementation of crossover operation are closely related to the researched problem. It is generally required that it is not only undermines the excellent pattern of the individual coded string but also effectively produces some better new individual patterns [19]. The main types of crossover operation are as follows:

1. Single-point crossover, also known as a simple intersection, refers to the random setting of a cross point in the individual coded string and then exchanging part of the chromosomes of two pairs of individuals at that point. The important feature of single-point crossover is that if the relationship between adjacent loci can provide better individual traits and higher individual fitness, the possibility of this single cross operation to destroy this individual character and reduce the individual fitness is minimal.
2. Double-point crossover or multi-point crossover

Double-point crossover means that two crossover points are randomly set up in the individual string, and then some gene exchanges are performed. The specific operation process of double-point crossover is that first, two crossover points are randomly set in each of the two individual coding strings. Then, some chromosomes are exchanged between the two intersections of two individuals. Generally, multi-point crossover operator is not used because it may destroy some good patterns. In fact, as the number of crossover points increases, the possibility of individual structure being destroyed is gradually increasing, so it is difficult to effectively save the better pattern. Thus, the performance of the genetic algorithm is affected [20].

3. Uniform crossover

Uniform crossover refers to the exchange of genes on each gene locus of two paired individuals with the same crossover probability, thus forming two new individuals. The

specific operation can be determined by setting up the shield character to determine which parent genes provide the new genes.

4. Arithmetic crossover

Arithmetical crossover refers to the creation of two new individuals by the linear combination of two individuals. In order to be able to perform linear combinatorial operations, the objects of arithmetic crossover are generally the individuals represented by floating-point numbers. Assume arithmetical crossover is executed on two individuals X_A^{t+1} and X_B^{t+1} , then two new individuals after crossover operation are expressed as

$$\begin{cases} X_A^{t+1} = \alpha X_B' + (1 - \alpha) X_A' \\ X_B^{t+1} = \alpha X_A' + (1 - \alpha) X_B' \end{cases} \quad (8)$$

where α is parameter. If it is a constant, this crossover operation is called uniform arithmetic crossover. If it is a variable determined by the number of evolutionary generations, this crossover operation is called nonuniform arithmetic crossover. The operation of the arithmetic crossover is to first determine the coefficient of the linear combination of individuals and then generate two new individuals according to Eq. (8).

3.3 Assignment of genetic operator of music creation

l is the length of code string. When binary code is used to represent the individual, the selection of the length of the encoding string Z is related to the solution accuracy of the problem. When floating-point code is used to represent individuals, the length of the encoding string l is equal to the number n of the decision variables. When using symbols to represent individuals, the encoding string length is determined by the encoding mode of the problem. In order to improve the local search ability of genetic algorithm, gray code can be used. In addition, variable length coding can be used to represent individuals.

M is the group size, that is, the number of individuals in the computer music creation population. When the M value is small, the operation speed of the genetic algorithm can be improved. But it will reduce the diversity of the population, which may cause premature convergence of the genetic algorithm. When the M value is large, the operation efficiency of the genetic algorithm will be reduced. It is generally set to 201,000.

T is the number of terminated evolutionary generation of genetic operations. It refers to that the genetic algorithm stops running after running to the specified evolutionary generation. It is generally set to 20~100.

P is the crossover probability. Crossover operation is the main method to generate new individuals in genetic algorithm, so the crossover probability should generally take a larger value. However, if the value is too large, it will destroy the good mode of the group and have adverse effects on the evolutionary computation. If the value is too small, the speed of producing new individuals is slower. It is generally set to 0.4~0.99.

Pm is mutation probability. If the mutation probability is large, although it can produce more new individuals, it may also destroy many good modes, which leads to the performance of genetic algorithm approximate to the performance of random search algorithm. If the mutation probability is too small, the ability of mutation operation to produce new individuals and the ability to suppress premature phenomena will be poor. It is generally set

to 0.0001~0.1. It is necessary to explain that the above operating parameters have a certain effect on the results and efficiency of solving the genetic algorithm in computer music creation, but there is no theoretical basis for them to be selected at present. In the practical application of genetic algorithm, it is often necessary to do several trial calculations before the reasonable value, or range of these parameters can be determined. The flow chart of the basic genetic algorithm is shown in Fig. 5.

4 Experiment

4.1 Determination of the similarity of computer music creation

Because the similarity of computer music creation is the basis for defining a cluster, the measurement of the similarity between the two patterns in the same feature space is

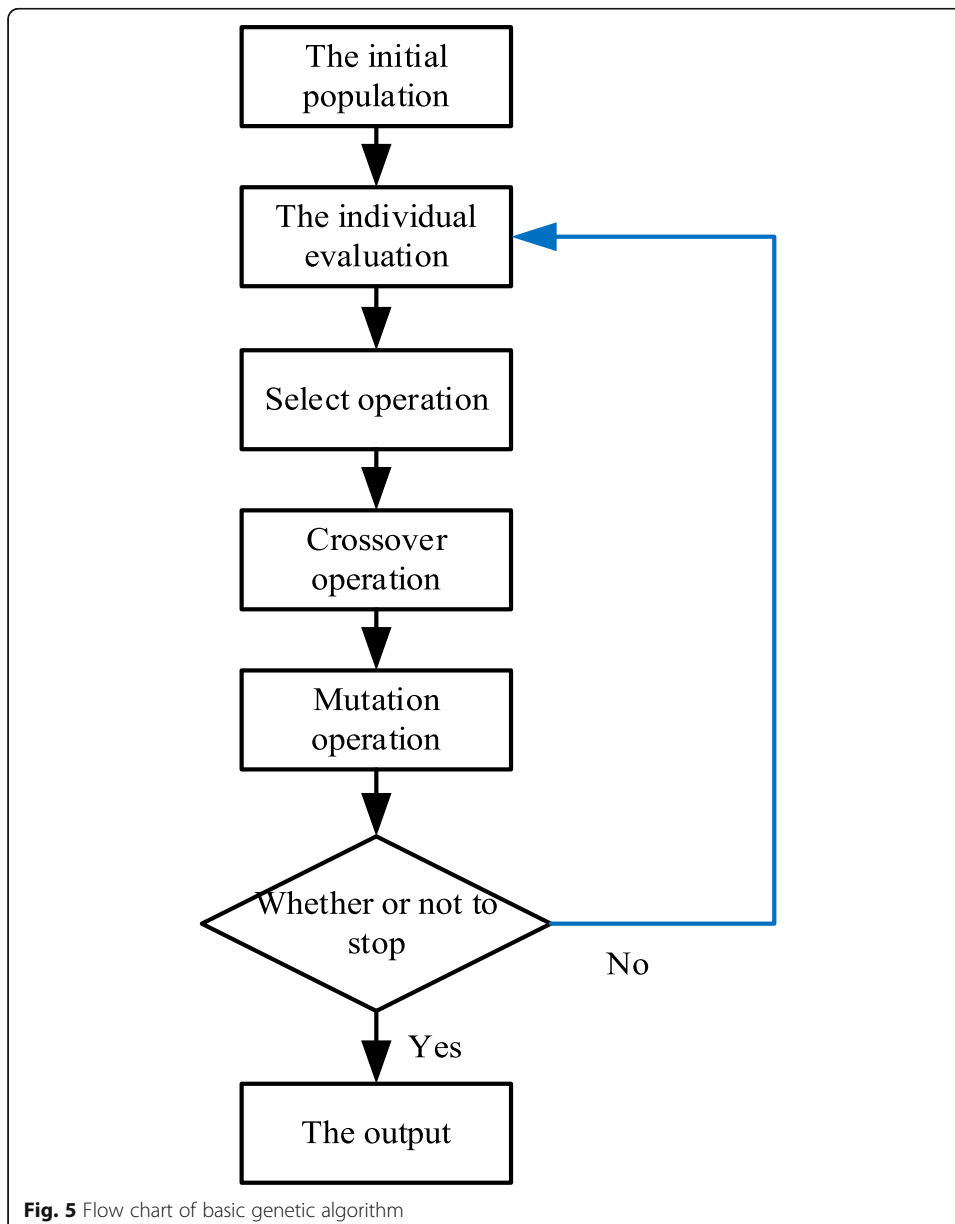


Fig. 5 Flow chart of basic genetic algorithm

essential to most clustering algorithms. In general, the similarity degree between two samples is not calculated directly, but the distance between the two samples is used to calculate the dissimilarity between the samples. For a sample space, the measurement standard of distance can be measured or semi-measured so as to quantify the dissimilarity of samples.

The measurement standard of the dissimilarity of the computer music creation is denoted as $d(x, x'), \forall x, x' \in X$. The dissimilarity is usually called distance. If x and x' are similar, the distance $d(x, x')$ is small. If x and x' are not similar, the distance $d(x, x')$ is large. Assume

$$d(x, x'), \forall x, x' \in X \quad (9)$$

The dissimilarity has symmetry and trigonometric inequality,

$$\begin{cases} d(x', x) = d(x, x'), \forall x, x' \in X \\ d(x, x'') \leq d(x', x) + d(x, x'), \forall x, x' \in X \end{cases} \quad (10)$$

Commonly used distance measurement standards are as follows. Assume the vector $x_i \in X, i = 1, 2, \dots, n, m$ is the number of dimensions; then, Euclidean distance of m -dimensional feature space is given by

$$d(x_i, x_j) = \sum_{k=1}^m [(x_{ik} - x_{jk})^2]^{\frac{1}{2}} \quad (11)$$

Manhattan distance of m -dimensional feature space is given by

$$d(x_i, x_j) = \sum_{k=1}^m |x_{ik} - x_{jk}| \quad (12)$$

Minkowski distance of m -dimensional feature space is given by

$$d(x_i, x_j) = \sum_{k=1}^m [(x_{ik} - x_{jk})^p]^{\frac{1}{p}} \quad (13)$$

4.2 Mean clustering analysis of computer music creation

Considering the relevance of computer music creation, the mean clustering method is adopted. Mean algorithm is widely used in clustering of large-scale datasets. The reasons for the popularity of mean algorithm in computer music creation are as follows:

Step 1. The time complexity is $O(nkl)$. n is the number of samples, k is the class number, and the l is the number of iterations when the algorithm converges. In general, k and Z are given in advance, so the time complexity of the algorithm is linear with the size of the dataset.

Step 2. The spatial complexity is $O(k + n)$. If you can store all the data in the main memory, access to all elements is very fast, and the efficiency of the algorithm is very high.

Step 3. It is an algorithm that does not depend on the order. Given an initial class distribution, regardless of the order of the samples, the data partitions generated after the partitioning process are the same.

In the computer music creation, mean clustering method is a partition clustering method based on variance criterion. The number of classes is fixed to generate a partition with the smallest variance. Assume n -dimensional sample space is partitioned into k classes by n modes $\{C_1, C_2, \dots, C_k\} (k = 1, 2, \dots, K)$. Each C includes an nk sample table. Then, the center of the class is given by

$$M_k = (1/n) \sum_{i=1}^m x_{ik} \quad (14)$$

where x_{ik} is the i th sample of the class C_k . The variance of C_k is the sum of squares of the Euclidean distance between each sample and its center. This error is called the between-class error, expressed as

$$e_k^2 = \sum_{i=1}^{nk} (x_{ik} - m_k)^2 \quad (15)$$

The squared error of the clustering space with k classes is the sum of between-class variance, given by

$$E_k^2 = \sum_{k=1}^k e_k^2 \quad (16)$$

The mean clustering algorithm for computer music creation is as follows. For given k , k partitions are to be found to let E_k^2 minimum. The mean clustering algorithm is as follows:

Step 1. Randomly generate k centers.

Step 2. Assign each point to the nearest center according to a certain distance measure.

Step 3. For each center, calculate the center of gravity of all points belonging to this center as new central coordinates.

Step 4. If a center changes, turn to step (2).

Step 5. End of calculation. Return to the k center position.

4.3 Completion of computer music creation

According to Eq. (5), the computer music creation with the constraints can be described as

$$\min f(x) = \left[-f_1(x), -f_2(x), \dots, -f_s(x), f_{s+1}(x), f_{s+2}(x), \dots, f_p(x) \right]^T \quad (17)$$

In practical application, when using genetic algorithm, the constraints of computer music creation must be processed. In general, inequality constraints are easy to operate by genetic algorithm, so the larger equality constraints are processed. The comparison of vectors is complex and time-consuming, so we can use vector norm to measure the length of vectors. The comparison of vectors is converted to the comparison of vector length, which can greatly improve the efficiency of the algorithm. Let

$$D_i(x) = \left(\sum_{i=1}^M [c_i(x)]^p \right)^{\frac{1}{p}} \quad (18)$$

$D_i(x)$ reflects the length of the feasible space for each individual. $D_i(x) = 0$ represents that the individual is located in the feasible space and is a feasible solution. $D_i(x) = D_j(x)$ represents the two individuals have the same length from the feasible space. $0 < D_i(x) < D_j(x)$ represents that although $D_i(x)$ and $D_j(x)$ are infeasible solutions, J individuals are closer to feasible space than i individuals, that is, J individuals are superior to i individuals. In Fig. 6, $D1$ is a feasible solution. Although $D2$ and $D3$ are not feasible solutions, $D2$ is better than $D3$ because $D2$ distance is closer to feasible space. $D(x)$ is called as a *D-fitness*. The mapping of search space to feasible space can be realized by using *D-fitness*. On the other hand, the whole group can be divided into feasible group and infeasible population. More importantly, *D-fitness* can be used as fitness of genetic algorithm, and a large number of feasible individuals can be generated through selection, crossover, and mutation.

The solution of multi-objective optimization genetic algorithm with complex constraints can be divided into infeasible solutions and feasible solutions, in which the infeasible individual sets constitute the infeasible group and feasible individual sets constitute the feasible population. For some highly constrained problems, a few feasible solutions may not be generated or generated in previous generations. If the Pareto operation and the mean clustering analysis are applied to the set of feasible solutions, the efficiency of the algorithm will be reduced, and there is not much practical significance. So, the first step of this algorithm is to generate a large number of feasible solutions. At this time, the *D-fitness* is used as the fitness of the genetic algorithm. Through the operations of selection, crossover, and mutation, the feasible solutions are generated in the previous generations, which is the basis for the Pareto operation.

According to the definition of Pareto optimal solution, if there exists an optimal solution in the multi-objective optimization problem, this solution must be Pareto optimal solution. So, the key to solve the multi-objective optimization problem is to find all the Pareto optimal solutions. In this algorithm, when the number of feasible solutions is sufficient, the Pareto operation can be carried out to the feasible population, and all the

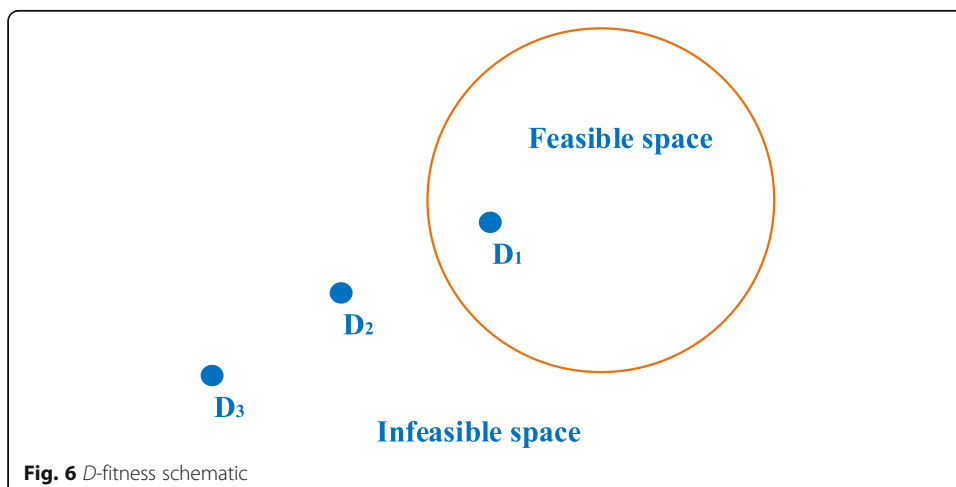


Fig. 6 *D-fitness* schematic

Pareto optimal solutions are found. Thus, the feasible group is divided into feasible non-Pareto group and feasible Pareto population.

The feasible Pareto group is divided into non-clustering Pareto group and clustering Pareto optimal population. Since the basic genetic algorithm is not limited to the number of the same or similar individuals, there may be a large number of same or similar individuals in a feasible Pareto population, especially in the late stage of the algorithm. Therefore, it is necessary to limit their numbers in order to generate more abundant Pareto optimal population. In this algorithm, the mean clustering analysis is used to achieve group diversity. The mean clustering analysis of feasible Pareto group is conducted, and q clustering populations are generated. An individual is randomly selected in each cluster group and merged into a population. This group is called clustering Pareto optimal population. The structure organization is shown in Fig. 7.

Finally, the whole group of computer music creation is classified into four groups: infeasible group, feasible non-Pareto group, non-clustering Pareto group, and clustering Pareto optimal group. The infeasible group is the worst adaptable group in the whole group. In order to reflect the optimal conservation strategy, the probability of their inheritance should be reduced to the next generation. Clustering Pareto optimal group is the most adaptable elite group selected from the feasible Pareto group. The probability of their inheritance to the next generation should be improved. To achieve this goal, R -fitness can be assigned to these four groups. Their relationship is as follows:

$$R(\text{infeasible group}) \ll R(\text{feasible non-Pareto group}) \ll R(\text{clustering Pareto group}) \ll R(\text{clustering Pareto optimal group})$$

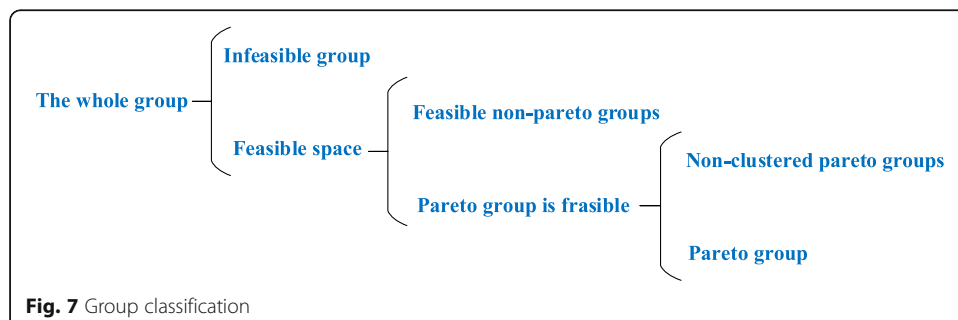
At this time, R -fitness can be used as the fitness of the algorithm, which is the basis for the selection operation of multi-objective optimization genetic algorithm.

The process of realizing computer music creation is as follows:

Step 1. Set the number of evolutionary generations $t \leftarrow 1$. The initial computer music creation group $P(t)$ with M initial individuals is generated.

Step 2. According to the constraint condition, the whole group is divided into infeasible group $P(t)$ and feasible group $p(t)'_1$ by D -fitness.

Step 3. The fitness values of each individual are calculated for the feasible computer music creation group $p(t)'_1$, and then Pareto operation is performed. The feasible group $p(t)'_1$ is divided into feasible non-Pareto group $P(t)2$ and feasible Pareto group $p(t)'_2$.



Step 4. For the feasible Pareto group $p(t)_2'$, the mean cluster analysis operation is performed. The feasible Pareto group $p(t)_2'$ is divided into non-clustering group $p(t)_3$ and clustering Pareto optimal group $p(t)_3'$.

Step 5. The appropriate *R-fitness* values are assigned to the four subgroups of $p(t)_1$, $p(t)_2$, $p(t)_3$, and $p(t)_3'$.

Step 6. The *R-fitness* is used to select proportion for computer music creation group.

Step 7. Single-point crossover operation is performed for computer music creation group.

Step 8. Uniform mutation operation is performed for computer music creation group.

Step 9. Judge the end condition. If the end condition is not satisfied, the updated generation counter $t \leftarrow 2$ is updated. If the end condition is satisfied, the computer music creation result is output, and the algorithm is end.

Step 10. The computer music creation is achieved.

5 Results

In order to ensure the validity of the proposed method, simulation and analysis is carried out. During the experiment, different types of music were used as the subjects to conduct the simulation test of syllable pitch accuracy. The different frequencies and types of music types are simulated. In order to ensure the effectiveness of the test, the conventional computer music creation method is taken as the comparison object, and the results of the two simulation experiments are compared.

5.1 Test data preparation

In order to ensure the accuracy of the simulation test process, the parameters of the test are set. In the process of test, it is necessary to ensure the consistency of test environment parameters. The form of computer music creation is shown in Table 1.

In order to ensure the rigor of the simulation process, two different computer music creation methods are selected, and the computers with the same performance are selected for music creation. The computer device environment is shown in Table 2.

Table 1 Computer music creation form

Type	Frequency change rate	Time
Waltz	0.9~2.1 kHz	3 min and 45 s
Concerto	1.2~5.4 kHz	4 min and 2 s
Symphony	1.8~2.4 kHz	5 min and 41 s
March	2.0~6.2 kHz	4 min and 14 s
Fantasia	2.5~5.0 kHz	5 min and 14 s
Serenade	3.0~6.1 kHz	6 min and 36 s
Sonata	3.7~7.0 kHz	7 min and 14 s
Variation	4.1~6.9 kHz	5 min and 48 s
Minuet	5.4~7.8 kHz	4 min and 49 s
Romantic	5.4~8.0 kHz	4 min and 19 s
Symphony	6.2~8.0 kHz	3 min and 42 s

Table 2 Computer device environment

Project	Type parameters	Version parameter
The operating system	Windows	7Ultimate
Central processing unit	Intel Core i7 (Bloomfield)	Four core eight threads Faster, 2.8 GHz Our firm frequency, 3.46 GHz
The sound card	Asus Xonar D - Kara	88.2 kHz, 24 bit
The graphics card	NVIDIA Rainbow iGame GTX 1080Ti Vulcan X OC	Core frequency, 1480/1733 mhz Display frequency, 11000 MHz Video memory capacity, 11 GB Display memory bit width, 352 bit
Memory chips	Ruida platinum winning sport LT 8GB DDR4 2400	Memory type, DDR4 Main memory frequency, 2400 MHz Memory capacity, 8 GB

5.2 Test process and result analysis

The syllables and tones of standard music are simulated during the experiment. Two different computer music creation methods are compared with simulated values to analyze the change of syllable pitch accuracy. The analysis results cannot be directly compared for two different methods of computer music creation. Therefore, the third-party analysis and recording software *Analysis* are adopted. The test process and result are recorded and analyzed. In the simulation test result curve, the *Analysis* function is used to eliminate the uncertainty caused by the factors of computer simulation and personnel operation. The simulation test of syllable pitch accuracy is carried out for different types of music and different methods of computer music creation. The syllabic accuracy test curve is shown in Fig. 8.

As shown in Fig. 8, it can be seen that, compared with the conventional method, the accuracy of syllable of the proposed computer music method based on genetic

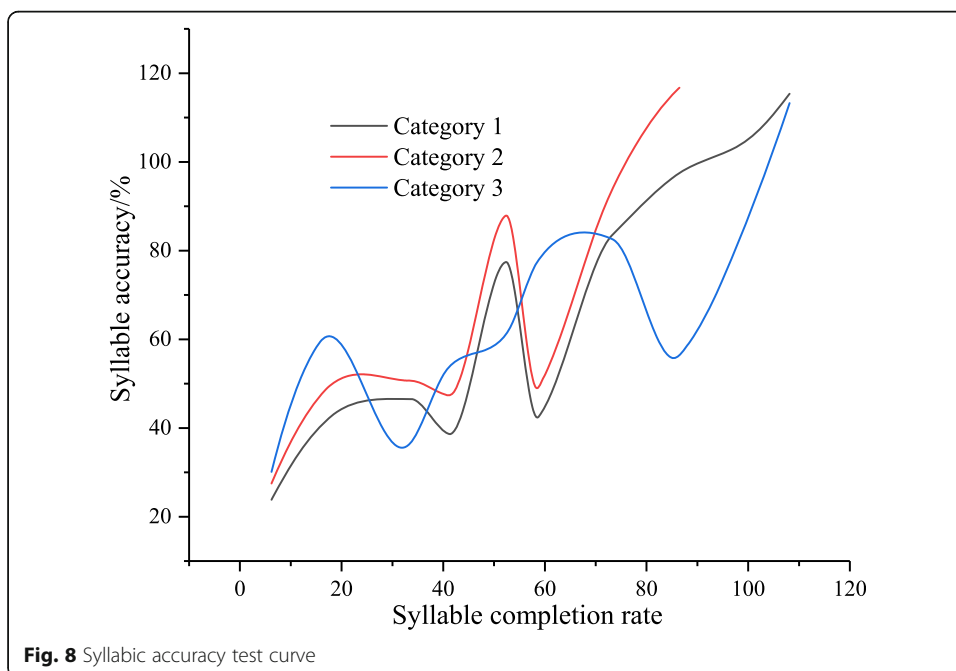


Fig. 8 Syllabic accuracy test curve

algorithm is increased by 25.97%. The tone change frequency of the proposed method is basically the same as the standard tone produced by the simulation. The pitch accuracy of two different computer music creation methods is tested. The pitch accuracy test curve is shown in Fig. 9.

As shown in Fig. 9, it can be seen that, compared with the conventional method, the pitch accuracy test of the proposed computer music method based on genetic algorithm is increased by 34.12%. The tone change frequency of the proposed method is basically the same as the standard tone produced by the simulation, and it can be used to create computer music with different transformation rhythm.

6 Discussion

This paper proposes a method of music creation and sound test based on genetic algorithm and wireless network. It aims to solve the problem that when the rhythm of computer-generated music changes greatly, the frequency will increase, which will cause the problem of oversaturation of the tone function and distortion of the syllable and pitch. First, the article establishes a general mathematical module for music creation. The best solution of Pareto is used to realize computer music creation. Thus, completed the research of computer music creation based on genetic algorithm. In addition, in order to ensure the effectiveness of the proposed computer music creation method, this paper simulates the music type test environment. Music audition work was carried out by a method based on efficient data transmission of wireless sensor network. Two different computer music creation methods are used to simulate syllable pitch accuracy. Experimental results show that the proposed computer music creation and sound test methods are effective.

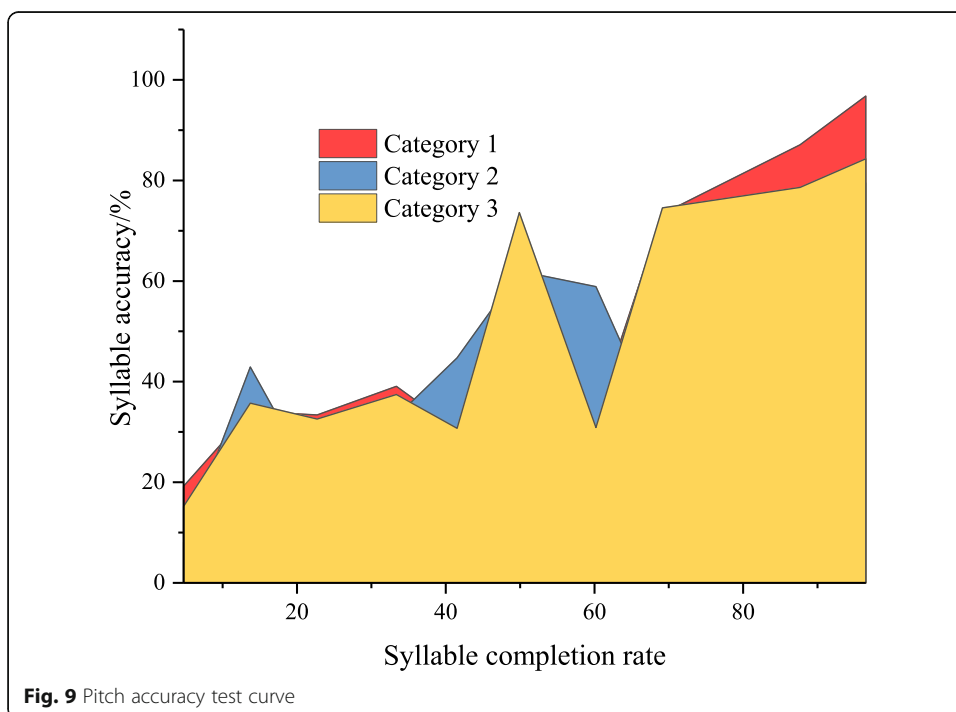


Fig. 9 Pitch accuracy test curve

Abbreviations

CCD: Charge Coupled Device

Acknowledgements

Key research projects of Humanities and Social Sciences in colleges and universities of Anhui Province in 2017: Research on The Protection, Inheritance and Innovation of Red Revolution Songs of National Intangible Cultural Heritage "Dabie Mountain Folk Songs (No. SK2017A0405)".

Authors' contributions

Lingzi Wang wrote the entire article. Mingxia Pan is responsible for processing the data. The authors read and approved the final manuscript.

Funding

Key research projects of Humanities and Social Sciences in colleges and universities of Anhui Province in 2017: Research on The Protection, Inheritance and Innovation of Red Revolution Songs of National Intangible Cultural Heritage "Dabie Mountain Folk Songs (No. SK2017A0405)".

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics approval and consent to participate

This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication

All authors agree to submit this version and claim that no part of this manuscript has been published or submitted elsewhere.

Competing interests

The authors declare that they have no conflict of interest.

Received: 3 April 2020 Accepted: 6 July 2020

Published online: 17 July 2020

References

1. Y. Liu, Music library retrieval feature tone song intelligent recognition simulation. *Comp. Simul* **34**(8), 356–359 (2017)
2. X.M. Fu, F.L. Wang, Optimization of BP neural network based on multi-offspring genetic algorithm. *Comp. Simul* **33**(3), 258–263 (2016)
3. Y.P. Wang, C. Guo, Genetic optimization control simulation of aircraft hydraulic servo system EHA. *Comp. Simul* **33**(11), 53–57 (2016)
4. Y. Lu, K.G. Jiang, Application of an improved genetic algorithm in TDOA localization. *Comp. Simul* **33**(12), 329–332 (2016)
5. H.S. Lang, S.C. Lin, Subcarrier pairing and power allocation with interference management in cognitive relay networks based on genetic algorithms. *IEEE Trans. Veh. Technol.* **65**(9), 7051–7063 (2016)
6. T. Xu, Y. Wang, C. Kang, Tailings saturation line prediction based on genetic algorithm and BP neural network. *J. Intell. Fuzzy Syst.* **30**(4), 1947–1955 (2016)
7. F.S. Structures, Inverse kinematic solutions of dual redundant camera robot based on genetic algorithm. *Math. Probl. Eng.* **1**, 1–10 (2017)
8. Z. Fu, Inverse determination of thermal conductivity in lumber based on genetic algorithms. *Holzforschung* **70**(3), 235–241 (2017)
9. M. Nemati, M. Braun, S. Tenbohlen, Optimization of unit commitment and economic dispatch in microgrids based on genetic algorithm and mixed integer linear programming. *Appl. Energy* **69**(3), 210–218 (2017)
10. K.H. Ibrahimi, S.M. Saleh, Design of multi-step LC compensator for time-varying nonlinear loads based on genetic algorithm. *Int. Transac. Elect Energy Syst.* **26**(12), 2643–2656 (2017)
11. H. Marshoud, H. Otok, H. Barada, Realistic framework for resource allocation in macro-femtocell networks based on genetic algorithm. *Telecommun. Syst.* **63**(1), 99–110 (2017)
12. I. Aouadni, A. Rebai, Decision support system based on genetic algorithm and multi-criteria satisfaction analysis (MUSA) method for measuring job satisfaction. *Ann. Oper. Res.* **256**(1), 3–20 (2017)
13. P. Renna, A decision investment model to design manufacturing systems based on a genetic algorithm and Monte-Carlo simulation. *Int. J. Comput. Integr. Manuf.* **30**(6), 590–605 (2017)
14. B.R. Ding, X.H. Qu, Y.L. Chen, Application and research of mechanical design optimization based on genetic algorithm kinematics simulation technology. *J. Intell. Fuzzy Syst.* **34**(2), 871–878 (2018)
15. L.Y. Wang, Y. Zhao, L.P. Li, Research on the vibration characteristics of the commercial-vehicle cabin based on experimental design and genetic algorithm. *J. Vibroengin.* **18**(7), 4664–4677 (2018)
16. A. Senoussi, S. Dautère-Pérés, N. Brahim, Heuristics based on genetic algorithms for the capacitated multi vehicle production distribution problem. *Comput. Oper. Res.* **34**(2), 871–878 (2018)
17. A. Filho, A.C. Silva, A.C. Paiva, Computer-aided diagnosis system for lung nodules based on computed tomography using shape analysis, a genetic algorithm, and SVM. *Med. Biol. Eng. Comput.* **55**(8), 1129–1146 (2016)

18. J.D. Guedes, D.D. Ferreira, A non-intrusive approach to classify electrical appliances based on higher-order statistics and genetic algorithm: a smart grid perspective. *Electr. Power Syst. Res.* **34**(2), 140,65–140,69 (2016)
19. M. Tayyebtaher, S.M. Esmailzadeh, Model predictive control of attitude maneuver of a geostationary flexible satellite based on genetic algorithm. *Adv. Space Res.* **55**(8), 1129–1146 (2017)
20. C. Liu, Y. Liu, Research on optimal allocation model and algorithm of radar interference resources. *Comp. Simul* **33**(5), 23–26 (2016)

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
