

# Automated Parameter Optimization for Double Frequency Modulation Synthesis Using the Genetic Annealing Algorithm\*

B. T. G. TAN AND S. M. LIM

*Department of Physics, National University of Singapore, Singapore 0511, Republic of Singapore*

A combinatorial optimization algorithm, called the genetic annealing algorithm, is proposed, which incorporates the principles of simulated annealing into the genetic algorithm, for the estimation of the parameters for double frequency modulation (DFM) synthesis, which is a tedious process by the usual trial-and-error method. It is also shown that the algorithm is more effective than the genetic algorithm in estimating the DFM parameters for all the authors' samples.

## 0 INTRODUCTION

The frequency modulation (FM) technique [1] made the generation of a wide range of musical sounds by real-time digital computation possible. The basic FM operator uses a sine wave (the modulator) to frequency modulate another sine wave (the carrier), producing a complex wave whose spectrum depends on the frequencies of the two sine waves and the degree of modulation, as given by the equation

$$x(t) = A(t) \sin[\omega_c t + I(t) \sin \omega_m t] \quad (1)$$

where  $A(t)$  is the time-varying amplitude,  $\omega_c$  and  $\omega_m$  are the carrier and modulator frequencies, respectively, and  $I(t)$  is the time-varying modulation index. More complex spectra can be generated by cascading or combining more than one operator.

A variant of the FM technique is asymmetrical FM (AFM) [2], which adds an additional parameter to the FM equation. This results in spectra that are asymmetrical about the carrier frequency, thus enabling a wider range of musical spectra to be synthesized. However, AFM makes much greater computational demands than FM. Another variant of FM that has been proposed, double frequency modulation (DFM) [3], also allows more complex spectral envelopes than FM to be generated, but without the computational load of AFM.

The DFM synthesis equation is

$$x(t) = A(t) \sin[I_1(t) \sin \omega_1 t + I_2(t) \sin \omega_2 t] \quad (2)$$

where  $A(t)$  is the time-varying amplitude and  $I_1(t)$  and  $I_2(t)$  are the time-varying modulation indices of the two modulator frequencies  $\omega_1$  and  $\omega_2$ , respectively.

Careful expansion [4] of Eq. (2) yields the following spectral equation:

$$x(t) = A(t) \sum_i \sum_k J_i(I_1(t)) J_k(I_2(t)) \sin(i\omega_1 t + k\omega_2 t) \quad (3)$$

where the amplitudes of the various harmonics are each determined by the product of two Bessel functions of the first kind. The indices in Eq. (3) run from 1 to  $+\infty$ .

The frequency bandwidth and the complexity of the DFM harmonic spectra are dependent on the ratio of the two modulator frequencies and the value of the modulation indices. As the indices increase, the amplitudes of the various harmonics vary in a nonlinear fashion, resulting in very complex spectra when the values of the indices are high. Hence, as in FM and AFM, various interesting sounds can be obtained from the DFM equation by varying the DFM parameters, namely, the two frequencies  $\omega_1$  and  $\omega_2$  and the two indices  $I_1$  and  $I_2$  in Eq. (2).

DFM synthesis is able to generate more complex harmonic envelopes than simple FM while avoiding the computational load of AFM synthesis. Due to its complexity, estimation of the parameters for DFM synthesis

\* Manuscript received 1995 June 6; revised December 5.

by trial and error is tedious. To speed up the process, we have used computer optimization techniques which search for the most suitable parameters for a given spectral envelope. Specifically, we propose an algorithm that incorporates simulated annealing [5] into a genetic algorithm [6], [7]. It has been shown [8] that FM parameter optimization using a genetic algorithm is much faster than the standard trial-and-error approach. We call this combination of simulated annealing and the genetic algorithm a genetic annealing algorithm. We have applied our genetic annealing algorithm to obtain an optimized set of parameters for DFM synthesis of a given musical spectrum. The results were compared with those obtained using a genetic algorithm. The combination of these two techniques gives a more optimized set of DFM parameters than if a genetic algorithm alone is used.

## 1 THE GENETIC ALGORITHM

The genetic algorithm is a search algorithm that was developed using the natural genetic processes by which organisms in nature evolve. The algorithm starts off with a random population of individuals whose characteristics are determined by a bit string, which has a function analogous to the chromosomes or genes in a biological individual. The "fitness" or "goodness" of each individual is defined in relation to a defined function—in nature this would be in terms of survival within a given environment, but for a particular optimization problem this would be determined by the nature of the desired solution. The bit strings are then evaluated one by one, and each is assigned a fitness or goodness value. The next generation of bit strings is then produced by a crossover process analogous to the natural combination of DNA from parents in reproduction. By this process, the fitter bit strings will be given a higher chance of being reproduced in the next generation. In this way, the average fitness of the next generation will improve, thus giving rise to better chromosomes. In addition, a mutation process may also occur, which can change bits in bit strings randomly, analogous to random mutation in nature.

The genetic algorithm thus consists of four steps:

- 1) Recruitment
- 2) Selection
- 3) Crossover
- 4) Mutation.

The recruitment process uses a random-number generator to generate sets of random parameters which will be encoded into individual bit strings to form a starting population. The fitness values of each individual are also evaluated. The selection process ensures that the fitter individuals are more likely to be selected than the less fit ones for the crossover process into the next generation. The crossover process mates the individuals in the current generation, using the weighting given by the selection process, to generate offspring for the next generation. The mutation process alters individual bits in the offspring randomly to simulate mutation of sites in a chromosome.

To apply the genetic algorithm to a combinatorial opti-

mization problem, each bit string has to represent the set of parameters that represent a solution to the problem. The goodness of each solution has to be defined by assigning a fitness function, determined by the nature of each problem, whose value is computed from the set of parameters. In the case of synthesis parameter optimization, the fitness is determined by the similarity of the harmonics generated by the synthesis to those of the waveform that is to be synthesized. The pseudo code of the genetic algorithm is as follows:

```

Begin
  RECRUITMENT
  Repeat
    SELECTION
    CROSSOVER PROCESS
    MUTATION
  Until GOOD_RESULTS
End

```

## 2 THE SIMULATED ANNEALING ALGORITHM

The simulated annealing algorithm is a probabilistic optimization algorithm which is based on the theory of annealing in condensed matter physics. Annealing is a thermal process that enables a solid to progress to a highly ordered crystalline form, which is its lowest energy state. In this process, the temperature of the solid is increased to its melting point in a heat bath, thus increasing the randomness of its atoms. The solid is then cooled very slowly to enable its atoms to reorder themselves so that it can reach a state of high order, that is, a low energy state with low internal strain. If the solid is not cooled sufficiently slowly, a strained amorphous structure may result. To prevent this, in the process of cooling the solid must be allowed to reach its thermal equilibrium at each temperature before any further cooling. Thermal equilibrium is characterized by a probability distribution of states with energy  $E$  given by the Boltzmann distribution,

$$P\{E\} = \frac{1}{Z(T)} \exp\left(-\frac{E}{k_B T}\right) \quad (4)$$

where  $T$  is the temperature of the solid,  $k_B$  is the Boltzmann constant, and  $Z(T)$  is the partition function, defined as

$$Z(T) = \sum_j \exp\left(\frac{E_j}{k_B T}\right) \quad (5)$$

which is the normalization factor in Eq. (4).

To model the physical annealing process using computer simulation, an analogy between the physical system and the optimization problem must be set up. The states of the physical system are analogous to the set of the independent variables or parameters of the system represented by the combinatorial optimization problem. The energy value of the physical system in a particular ordering of its atoms is analogous to the value of the goodness or fitness function of the system. Lowering

the energy value is analogous to obtaining a better value for the fitness function and hence getting a better solution. The aim is to get the best possible solution within the solution space of the problem, just as physical annealing puts the physical system in the lowest possible and most ordered energy state.

Thus the simulated annealing process based on physical annealing, as applied to combinatorial optimization, can be divided into the following steps:

- 1) Choose a sequence  $(T_k, t_k)$ , a value of  $K$ , and an initial state.
- 2) Perturb the initial state to a random neighboring state.
- 3) Perform a fitness test on the new state.
- 4) Repeat steps 2 and 3  $t_k$  times.
- 5) Repeat step 4  $K$  times.

Here  $(T_k, t_k)$  is a sequence of pairs representing, respectively, the temperature and the number of times of perturbation required at the  $k$ th generation of annealing, and  $K$  represents the total number of generations used for annealing. To perturb the current state, that is, step 2, a generation mechanism for the transition would have to be defined for the particular optimization problem. Step 3 refers to a fitness test in which, if the perturbed state is a fitter state than the current state, the perturbed state is accepted as the next current state. Otherwise the probability of acceptance of the perturbed state is given by Eq. (4). The pseudo code for the simulated annealing algorithm is as follows, where  $f(i)$  is the energy when the system is in state  $i$ :

```
BEGIN
  INITIALIZE;
  Repeat
    Repeat
      PERTURB(state i → state j);
      if  $f(j) < f(i)$  then  $i := j$ 
      else
        if  $\exp(f(i) - f(j)/T_k) > \text{Random}[0, 1]$  then UPDATE( $i := j$ );
    Until NEAR_THERMAL_EQUILIBRIUM;
  DECREMENT_TEMP ( $T_k$ );
  Until SOLID_FROZEN;
End
```

### 3 THE GENETIC ANNEALING ALGORITHM

In the genetic algorithm the crossover process is fundamental since it is the mechanism that generates new solutions which are offspring of the fitter solutions and thus more likely to be an improvement on the previous generation. However, the crossover mechanism is inherently a disruptive process as far as searching for the correct or fittest solution is concerned, since an offspring of two parents is not likely to be close to its parents in the solution space. Hence the genetic algorithm, while it can quickly arrive at better solutions through several generations, does not always converge to an optimum solution, which may be close to the parents.

On the other hand, the nature of simulated annealing is more localized in the sense that only a little perturba-

tion is performed to produce the next generation of solutions, often resulting in only small energy changes after perturbation. Hence, while simulated annealing has a greater tendency to converge to an optimum solution once nearby solutions have been reached, the convergence may take a longer time.

We therefore propose a new optimization algorithm, which we call the genetic annealing algorithm, which incorporates simulated annealing into the crossover process of the genetic algorithm and combines the advantages of both algorithms. In the genetic algorithm, the crossover function selects two individuals at random from the population to mate to produce offspring. In our genetic annealing algorithm, the crossover function is replaced by a new function, called the Anneal\_Cross function, where one of the parents is the fittest individual in the current generation whereas the mating partner is selected randomly. The two parents are mated randomly to produce offspring which will then be evaluated. The good ones will replace the parents for the next round of crossovers, whereas those that are less fit than their parents will only replace the parents with a certain probability characterized by the Boltzmann distribution. The mating will stop after a predefined number of offspring are produced. Another mating partner from the original population is then selected at random, and the mating process described repeats. The Anneal\_Cross function will stop when the number of partners mated reaches a predefined number. In effect, the Anneal\_Cross function incorporates a complete simulation annealing cycle into the crossover process.

Thus the Anneal\_Cross function can be divided into the following steps, in analogy with simulated annealing:

- 1) Choose sequence  $(T_k, t_k)$ , a value of  $K$ , and select the best individual.
- 2) Do crossover with a random partner.
- 3) Perform fitness test using the Boltzmann distribution.
- 4) Repeat steps 2 and 3 for a total of  $t_k$  times with the same partner.
- 5) Repeat steps 2, 3, and 4 with the new random partner for a total of  $K$  times.

$T_k$  is now called the fitness temperature,  $t_k$  is the number of crossovers with the  $k$ th partner, and  $K$  represents the total number of partners required for crossover. The crossing over in step 2 is equivalent to the crossover function defined in the simple genetic algorithm. Step 3 refers to a fitness test in which, if the offspring are fitter than the parents, then the former are accepted as the parents. Otherwise the probability of acceptance of the offspring is given by Eq. (4).

The pseudo code of the algorithm is given as follows:

• *Genetic annealing algorithm:*

```
Begin
  RECRUITMENT
  Repeat
    SELECTION
    ANNEAL_CROSS
  Until GOOD_RESULTS
End
```

• *Procedure Anneal\_Cross:*

```

Begin
  SELECT_FITTEST_INDIVIDUAL
  Repeat
    SELECT_RANDOM_MATING_PARTNER
    Repeat
      GENERATE_RANDOM_OFFSPRING
      if (OFFSPRING better than PARENTS)
        then UPDATE (PARENTS:= OFFSPRING)
      else UPDATE with Prob(Boltzmann Distribution)
    Until TIME_TO_CHANGE_PARTNER
  Until MAX_NO._OF_PARTNERS
End

```

#### 4 PARAMETER ESTIMATION PROCESS

The process by which the genetic algorithm or the genetic annealing algorithm is used to obtain an optimized set of parameters for the DFM synthesis of a particular musical instrument waveform consists of three steps.

The first step is the spectrum analysis process in which a sampled waveform of the musical instrument obtained from the McGill University Master Samples (MUMS) compact discs is analyzed on a Hewlett Packard dynamic signal analyzer using the fast Fourier transform (FFT). In the second step the genetic algorithm or the genetic annealing algorithm is used to obtain an optimized set of DFM parameters, using the spectrum from the signal analyzer to determine the fitness function by which the solutions generated are tested. Finally the optimized DFM parameters are input into the Motorola DSP56000 digital signal processor (DSP) to obtain the DFM synthesized sound of the musical instrument.

For the estimation and optimization of the DFM parameters, both algorithms begin with a recruitment process, as shown in Fig. 1. We will use the parameter  $f$  for frequency instead of  $\omega$ , where  $f = \omega/2\pi$ . In this process we use a uniform random generator to generate four random numbers standing for the four parameters for a single DFM operator, namely, the frequencies  $f_1$ ,  $f_2$  and the modulation indices  $I_1$ ,  $I_2$  with some user-defined constraints. The values of  $f_1$  and  $f_2$  have to be checked to ensure that they will generate the required harmonics, and the maximum value of the indices has to be defined depending on the characteristics of the sample spectrum. One set of four parameters represents one individual solution if only one DFM operator is used. For two DFM operators, one individual requires eight parameters. The number of individuals required to be generated will depend on the predefined population size. Each set of parameters generated is tested by being input into a fitness evaluation function which then assigns a fitness value to that individual. In our algorithm we have computed the fitness value by summing the squares of the absolute differences between the amplitudes of the corresponding harmonics of the real sample and those of the waveform synthesized from the given parameters.

After the recruitment process, a selection has to be

done based on the fitness values of the individuals. We choose to use the binary tournament selection scheme [8], as shown in Fig. 2, in which all individuals of the population are paired up to have their fitness values compared. The fitter individual of each pair is retained and the less fit individual discarded after comparison. This process (binary tournament) is repeated so that the new population size is equal to the old population size. This selection scheme will ensure that the best individual will appear twice in the new population and the worse individual will not be propagated at all.

After the binary tournament selection, the next process for the genetic algorithm is the crossover process followed by mutation, whereas that for the genetic annealing algorithm is the Anneal\_Cross process.

In both algorithms each individual bit string consists of four shorter strings of binary numbers, each of which represents one of the four parameters for one DFM operator. This representation ensures that at each crossing over, only one parameter of the four is crossed, resulting in the offspring being different only for that parameter. The other three parameters will remain the same after crossing. In this way, at each crossing the optimization process will work on only one parameter without affecting the other three parameters. We can also choose to keep one or more of the parameters constant throughout by restricting the crossover points to the parameters that are allowed to be varied. For example, we may choose to vary just the DFM indices and keep the frequencies constant, as we can still generate a wide variety of spectra in this way.

The crossover process is illustrated in Fig. 3. ParameterSelected in Fig. 3 refers to the parameter to cross, that

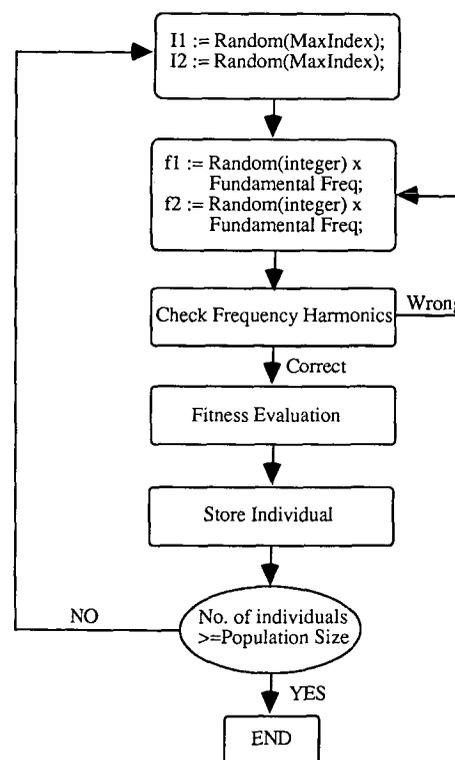


Fig. 1. Recruitment process.

is, whether to cross  $I_1$  or  $I_2$  or  $f_1$  or  $f_2$ . It is equal to a random integer number from 0 to 3, with each number representing one of the DFM parameters. Each of the parameter strings is 16 bits long, the bits being numbered from 0 to 15. Crossover Point is a random integer from 4 to 15. The bits less significant than this crossover point will be swapped between the selected parameter of Parent A and that of Parent\_B. For example, let the selected parameter be  $I_1$  and the crossover point be 8. If  $I_1$  of Parent\_A in binary is 0001101101011000 and  $I_1$  of Parent\_B is 0000010111011100, then the following are the values of  $I_1$  before and after crossing:

Parent_A. $I_1$ (before crossover)	00011011 01011000 <sub>b</sub>	or	1B58 <sub>h</sub>
Parent_B. $I_1$ (before crossover)	00000101 11011100 <sub>b</sub>	or	05DC <sub>h</sub>
Offspring_A. $I_1$ (after crossover)	00011011 11011100 <sub>b</sub>	or	1BDC <sub>h</sub>
Offspring_B. $I_1$ (after crossover)	00000101 01011000 <sub>b</sub>	or	0558 <sub>h</sub>

After the crossover process, mutation follows with 5% probability to change a random bit of the offspring. Using the new value of the parameter  $I_1$  together with the original values of the other parameters of the parents, the new fitness values for both can be evaluated. The offspring will then replace the parents, and the processes of selection, crossover, and mutation for the next generation will repeat until the offspring achieve the best possible optimized values.

For the genetic annealing algorithm, the new population has to undergo a mating or crossing process after the binary tournament selection, which we call the Anceal\_Cross process (Fig. 4) in our genetic annealing algorithm. To begin with, the best, that is, the fittest individual in the new population, which we call Parent\_A, is selected to be one of the parents for crossing. This best individual is allowed to mate with a partner, called Parent\_B, selected randomly from the population, to produce two offspring. An initial temperature is defined, in order that the thermal equilibrium can be defined according to a Boltzmann distribution function.

The two parents then undergo a crossover process similar to that for the standard genetic algorithm de-

scribed earlier to produce two offspring. Using the new value of the parameter  $I_1$  together with the other original values of the parameters of the parents, the new fitness values for both offspring can be evaluated. If the new fitness value of Offspring\_A is lower (that is, better) than the old fitness value of Parent\_A, then the parameter value of  $I_1$  will be updated to the new value. Otherwise the probability characterized by Boltzmann distribution is evaluated and compared with a normalized random number. The new value, which is less fit, is only accepted if the random number is less than the

calculated probability.

This mating process with the current partner is repeated with the two offspring taking the role of parents, generating a new pair of offspring, using different values of the selected parameter and the crossover point. This process is repeated for a specified number of times, sufficient to ensure that the best possible solution for the starting pair of parents is achieved. This is analogous to achieving thermal equilibrium in simulated annealing at a given temperature.

After this entire process, both Parent\_A and Parent\_B may or may not have been replaced by fitter individuals. The fittest individual from among the entire population is then selected. This may in fact be the same as the previous Parent\_A. Another partner will then be selected randomly from the population as Parent\_B to repeat the entire crossing process again. The temperature is also reduced by 10% for the purpose of the Boltzmann distribution comparison. In this way the best individual will have the chance to mate with different partners,

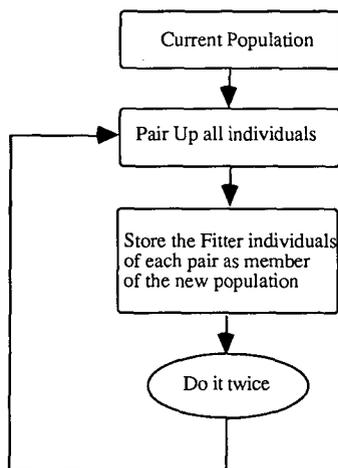


Fig. 2. Binary tournament selection process.

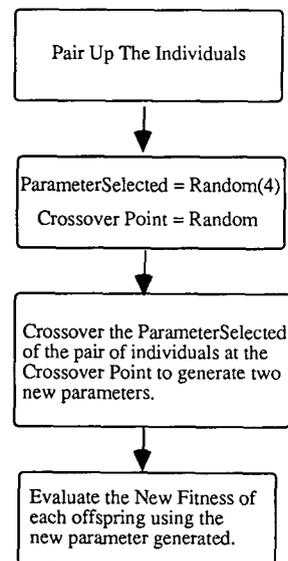


Fig. 3. Crossover process.

and for each partner the crossover is combined with a simulated annealing type procedure, as described.

The crossing over with a number of partners exploits the ability of the genetic algorithm to combine the good qualities of two different parents, while the simulated annealinglike procedure exploits its ability to search for optimal solutions over the whole solution space. We acknowledge that this is certainly not the only way in which the genetic algorithm can be combined with simulated annealing. Brown et al. [9] also combined simulated annealing with the genetic algorithm to solve the quadratic assignment problem, but they cascade both techniques, whereas in our genetic annealing algorithm the simulated annealing process is more closely integrated into the crossover procedure. In our case the computational load is probably less, which has contributed to the speed and quality of the optimized solution using our genetic annealing algorithm. Some optimized results obtained from a genetic algorithm and a genetic annealing algorithm are shown for comparison in the next section.

### 5 SYNTHESIS RESULTS

The FFT spectra of the real samples were obtained from the steady-state waveforms of each musical instrument from the MUMs compact disc set. Here we present the DFM synthesized spectra of a number of musical instruments. The parameter estimation was done independently by both the genetic algorithm and the genetic annealing algorithm. Their results are compared in Sec-

tions 5.1–5.4. The weight  $W[n]$  is used for the amplitude of each of the DFM operators, that is,

$$\begin{aligned}
 X(t) = & W[1] \sin[I_1[1] \sin(2\pi f_1[1]t) \\
 & + I_2[1] \sin(2\pi f_2[1]t)] \\
 & + W[2] \sin[I_1[2] \sin(2\pi f_1[2]t) \\
 & + I_2[2] \sin(2\pi f_2[2]t)]
 \end{aligned}$$

for the case of two DFM operators.

#### 5.1 The Oboe

The DFM parameters of an oboe were estimated to good accuracy using only two DFM operators. The spectra are shown in Fig. 5.

The parameters estimated by the genetic algorithm for the oboe are  $I_1 = 2.605$ ,  $f_1 = 660$ ,  $I_2 = 0.097$ ,  $f_2 = 880$  for the first operator, which carries a weight of 0.648809, whereas for the second operator,  $I_1 = 1.437$ ,  $f_1 = 440$ ,  $I_2 = 0.037$ ,  $f_2 = 220$ , carrying a weight of 1.849750. Since our fitness value is defined to be the sum of the square of the differences in the amplitudes of the sample and synthesized harmonics (as defined in Section 4), the lower the fitness, the better is the set of parameters. The fitness value of the resulting parameters, estimated by the genetic algorithm, is 0.047810.

By using the genetic annealing algorithm, the parameters estimated for the oboe are  $I_1 = 1.618$ ,  $f_1 = 660$ ,  $I_2 = 0.961$ ,  $f_2 = 880$  for the first operator, which carries

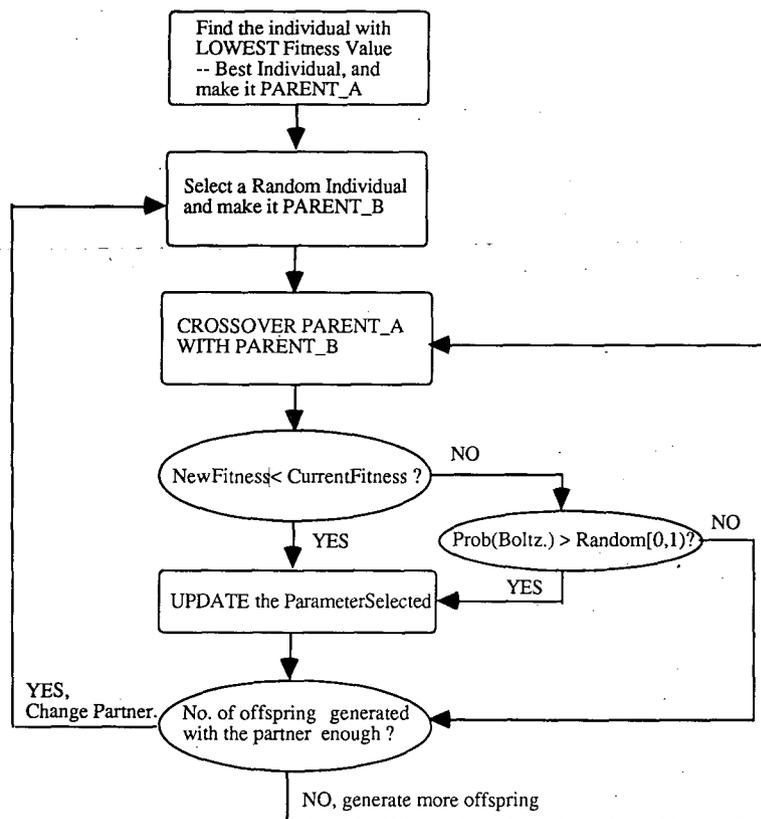


Fig. 4. Anneal\_Cross process.

a weight of 0.776809, whereas for the second operator,  $I_1 = 1.44, f_1 = 440, I_2 = 0.118, f_2 = 220$ , contributing a weight of 1.977013. The fitness value calculated for this set of parameters is 0.018030, which is much better than that of the genetic algorithm.

**5.2 The French Horn**

The DFM parameters of a French horn can be estimated to high accuracy using only two DFM operators. The spectra are shown in Fig. 6.

The parameters estimated by the genetic algorithm for the French horn are  $I_1 = 0.412, f_1 = 440, I_2 = 3.206641, f_2 = 220$  for the first operator, which carries a weight of 2.804504, whereas for the second operator,  $I_1 = 0.185, f_1 = 220, I_2 = 3.91, f_2 = 440$ , carrying a weight of -0.202384. The fitness value of the resulting parameters, estimated by the genetic algorithm, is

0.020239.

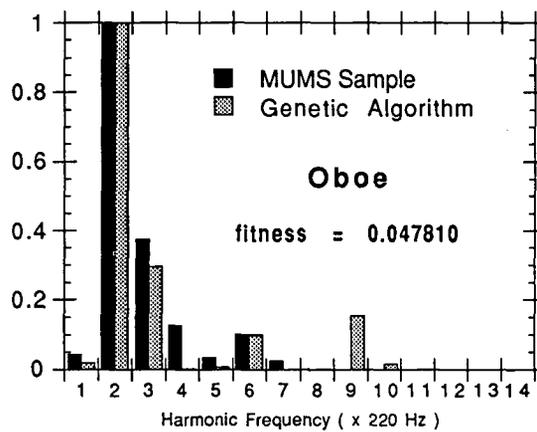
By using the genetic annealing algorithm, the parameters estimated for the French horn are  $I_1 = 0.384, f_1 = 440, I_2 = 3.174, f_2 = 220$  for the first operator, which carries a weight of 2.826249, whereas for the second operator,  $I_1 = 5.475, f_1 = 220, I_2 = 3.055, f_2 = 440$ , contributing a weight of -0.02898. The fitness value calculated for this set of parameters is 0.014579, which is again better than that of the genetic algorithm.

**5.3 The Saxophone**

The DFM parameters of a saxophone can be estimated to good accuracy using three DFM operators. The spectra are shown in Fig. 7.

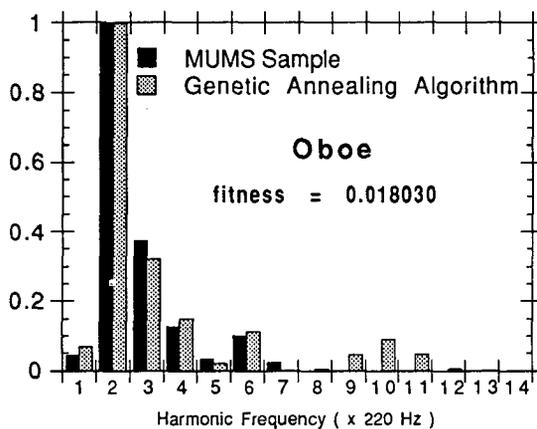
The parameters estimated by the genetic algorithm for the saxophone are  $I_1 = 1.470382, f_1 = 440, I_2 = 0.032411, f_2 = 220$  for the first operator, which carries

DFM OPERATOR A		DFM OPERATOR B	
AmpA	= 0.648809	AmpB	= 1.849750
IndexA1	= 2.605	IndexB1	= 1.437
FreqA1	= 660 Hz	FreqB1	= 440 Hz
IndexA2	= 0.097	IndexB2	= 0.037
FreqA2	= 880 Hz	FreqB2	= 220 Hz



(a)

DFM OPERATOR A		DFM OPERATOR B	
AmpA	= 0.776809	AmpB	= 1.977013
IndexA1	= 1.618000	IndexB1	= 1.440000
FreqA1	= 660 Hz	FreqB1	= 440 Hz
IndexA2	= 0.961000	IndexB2	= 0.118000
FreqA2	= 880 Hz	FreqB2	= 220 Hz



(b)

Fig. 5. DFM parameters of oboe. (a) Using genetic algorithm. (b) Using genetic annealing algorithm.

a weight of 2.204397, for the second operator,  $I_1 = 0.494217, f_1 = 440, I_2 = 3.222, f_2 = 220$ , carrying a weight of 1.801916, and for the third operator of weight  $-0.239075, I_1 = 2.788232, f_1 = 660, I_2 = 4.895, f_2 = 440$ . The fitness value of the resulting parameters, estimated by the genetic algorithm, is 0.024215.

By using the genetic annealing algorithm, the parameters estimated for the saxophone are  $I_1 = 5.055, f_1 = 220, I_2 = 2.073, f_2 = 440$  for the first operator, which carries a weight of 0.525090, whereas for the second operator,  $I_1 = 4.222, f_1 = 220, I_2 = 0.526, f_2 = 440$ , contributing a weight of 1.277130, and for the third,  $I_1 = 0.718, f_1 = 440, I_2 = 0.355, f_2 = 220$ , having a weight of 4.190951. The fitness value calculated for this set of parameters is 0.003901, which is still better than that of the genetic algorithm.

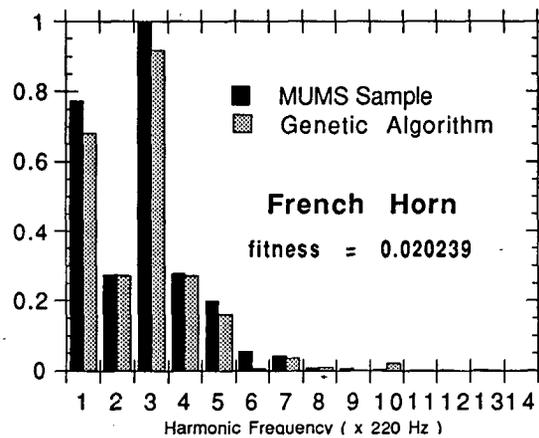
### 5.4 The Harpsichord

The DFM parameters of a harpsichord can be estimated to high accuracy using three DFM operators. The spectra are shown in Fig. 8.

The parameters estimated by the genetic algorithm for the harpsichord are  $I_1 = 0.29078, f_1 = 220, I_2 = 1.649, f_2 = 440$  for the first operator, which carries a weight of 1.949715, for the second operator,  $I_1 = 2.997, f_1 = 0.220, I_2 = 0.464, f_2 = 440$ , carrying a weight of 0.826759, and for the third operator of weight 0.608628,  $I_1 = 4.778222, f_1 = 220, I_2 = 0.129, f_2 = 440$ . The fitness value of the resulting parameters, estimated by the genetic algorithm, is 0.014251.

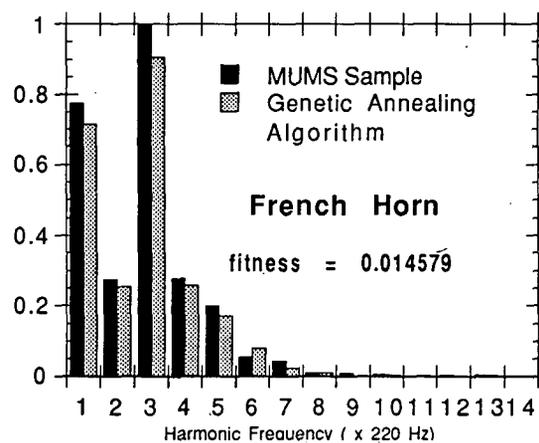
By using the genetic annealing algorithm, the parameters estimated for the harpsichord are  $I_1 = 2.367, f_1 =$

DFM OPERATOR A		DFM OPERATOR B	
AmpA	= 2.804504	AmpB	= -0.202384
IndexA1	= 0.412000	IndexB1	= 0.185000
FreqA1	= 440 Hz	FreqB1	= 220 Hz
IndexA2	= 3.206641	IndexB2	= 3.910000
FreqA2	= 220 Hz	FreqB2	= 440 Hz



(a)

DFM OPERATOR A		DFM OPERATOR B	
AmpA	= 2.826249	AmpB	= -0.028980
IndexA1	= 0.384000	IndexB1	= 5.475000
FreqA1	= 440 Hz	FreqB1	= 220 Hz
IndexA2	= 3.174000	IndexB2	= 3.055000
FreqA2	= 220 Hz	FreqB2	= 440 Hz



(b)

Fig. 6. DFM parameters of French horn. (a) Using genetic algorithm. (b) Using genetic annealing algorithm.

220,  $I_2 = 1.607$ ,  $f_2 = 440$  for the first operator, which carries a weight of 0.546932, whereas for the second operator,  $I_1 = 4.377$ ,  $f_1 = 220$ ,  $I_2 = 0.509$ ,  $f_2 = 440$ , contributing a weight of 1.056423, and for the third,  $I_1 = 0.745$ ,  $f_1 = 440$ ,  $I_2 = 0.189$ ,  $f_2 = 220$ , having a weight of 3.433636. The fitness value calculated for this set of parameters is 0.001890, which is again much better than that of the genetic algorithm. Thus the DFM parameters estimated by the genetic annealing algorithm can generate a spectrum that is much closer to the real sample of the harpsichord than those using the genetic algorithm.

### 5.5 The Trumpet

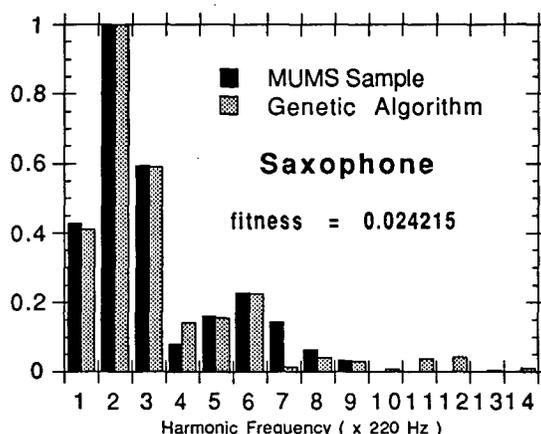
The DFM parameters of a trumpet can be estimated to good accuracy using three DFM operators. The spec-

tra are shown in Fig. 9.

The parameters estimated by the genetic algorithm for the steady-state trumpet are  $I_1 = 2.503$ ,  $f_1 = 440$ ,  $I_2 = 5.722$ ,  $f_2 = 220$  for the first operator, which carries a weight of 2.948724, for the second operator,  $I_1 = 0.019$ ,  $f_1 = 440$ ,  $I_2 = 5.238$ ,  $f_2 = 220$ , carrying a weight of 3.374904, and for the third operator of weight 5.366608,  $I_1 = 1.535$ ,  $f_1 = 440$ ,  $I_2 = 1.939695$ ,  $f_2 = 220$ . The fitness value of the resulting parameters, estimated by the genetic algorithm, is 0.684556.

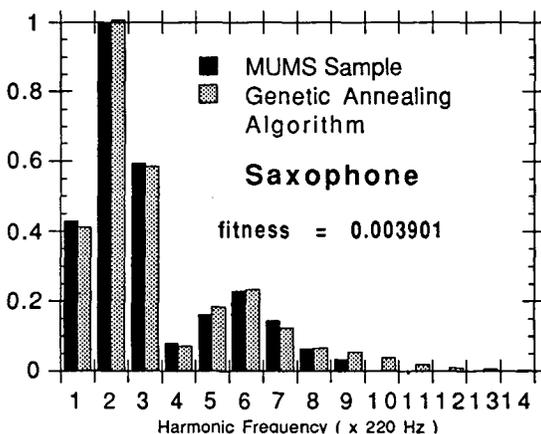
By using the genetic annealing algorithm, the parameters estimated for the trumpet are  $I_1 = 1.39$ ,  $f_1 = 220$ ,  $I_2 = 1.288$ ,  $f_2 = 440$  for the first operator, which carries a weight of 5.262182, whereas for the second operator,  $I_1 = 4.835$ ,  $f_1 = 220$ ,  $I_2 = 3.326$ ,  $f_2 = 440$ , contributing a weight of 2.815760, and for the third,  $I_1 = 5.246$ ,

DFM OPERATOR A	DFM OPERATOR B	DFM OPERATOR C
AmpA = 2.204397	AmpB = 1.801916	AmpC = -0.239075
IndexA1 = 1.470382	IndexB1 = 0.494217	IndexC1 = 2.788232
FreqA1 = 440 Hz	FreqB1 = 440 Hz	FreqC1 = 660 Hz
IndexA2 = 0.032411	IndexB2 = 3.222000	IndexC2 = 4.895000
FreqA2 = 220 Hz	FreqB2 = 220 Hz	FreqC2 = 440 Hz



(a)

DFM OPERATOR A	DFM OPERATOR B	DFM OPERATOR C
AmpA = 0.525090	AmpB = 1.277130	AmpC = 4.190951
IndexA1 = 5.055000	IndexB1 = 4.222000	IndexC1 = 0.718000
FreqA1 = 220 Hz	FreqB1 = 220 Hz	FreqC1 = 440 Hz
IndexA2 = 2.073000	IndexB2 = 0.526000	IndexC2 = 0.355000
FreqA2 = 440 Hz	FreqB2 = 440 Hz	FreqC2 = 220 Hz



(b)

Fig. 7. DFM parameters of saxophone. (a) Using genetic algorithm. (b) Using genetic annealing algorithm.

$f_1 = 220$ ,  $I_2 = 0.479$ ,  $f_2 = 440$ , having a weight of 4.787540. The fitness value calculated for this set of parameters is 0.155515, which is again much better than that of the genetic algorithm. Thus the DFM parameters estimated by the genetic annealing algorithm can generate a spectrum that is much closer to the real sample of the trumpet than those using the genetic algorithm.

### 5.6 The Pipe Organ

The DFM parameters of a pipe organ can be estimated to good accuracy using three DFM operators. The spectra are shown in Fig. 10.

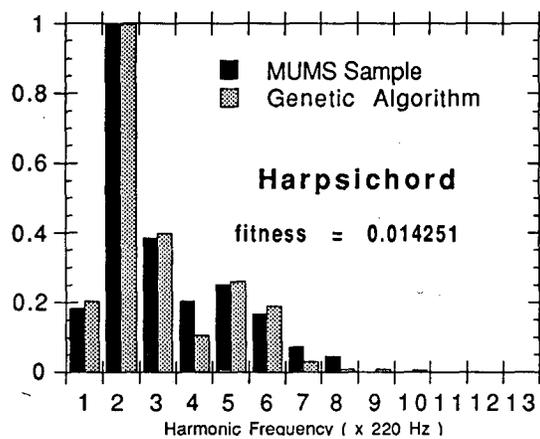
The parameters estimated by the genetic algorithm for the steady-state pipe organ are  $I_1 = 2.04$ ,  $f_1 = 220$ ,  $I_2 = 4.606$ ,  $f_2 = 440$  for the first operator, which carries a weight of 1.584622, for the second operator,  $I_1 =$

0.443,  $f_1 = 440$ ,  $I_2 = 3.024$ ,  $f_2 = 220$ , carrying a weight of 3.929953, and for the third operator of weight 2.219697,  $I_1 = 0.953917$ ,  $f_1 = 220$ ,  $I_2 = 2.014$ ,  $f_2 = 440$ . The fitness value of the resulting parameters, estimated by the genetic algorithm, is 0.582676.

By using the genetic annealing algorithm, the parameters estimated for the pipe organ are  $I_1 = 4.587$ ,  $f_1 = 440$ ,  $I_2 = 2.814$ ,  $f_2 = 220$  for the first operator, which carries a weight of 1.653801, whereas for the second operator,  $I_1 = 0.145$ ,  $f_1 = 220$ ,  $I_2 = 2.896$ ,  $f_2 = 440$ , contributing a weight of 2.743604, and for the third,  $I_1 = 0.774$ ,  $f_1 = 440$ ,  $I_2 = 2.922$ ,  $f_2 = 220$ , having a weight of 4.337077. The fitness value calculated for this set of parameters is 0.186845, which is again much better than that of the genetic algorithm.

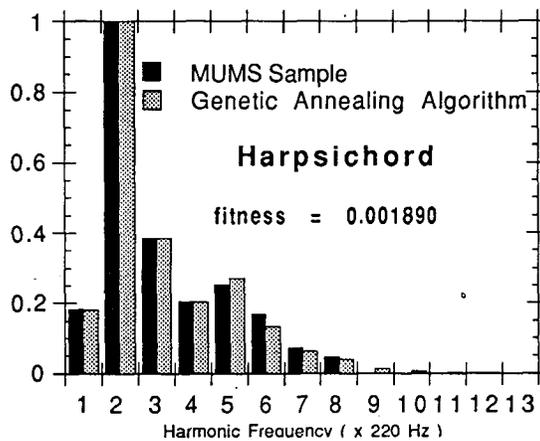
For all six instruments it is clear that the DFM param-

DFM OPERATOR A	DFM OPERATOR B	DFM OPERATOR C
AmpA = 1.949715	AmpB = 0.826759	AmpC = 0.608628
IndexA1 = 0.290780	IndexB1 = 2.997000	IndexC1 = 4.778222
FreqA1 = 220 Hz	FreqB1 = 220 Hz	FreqC1 = 220 Hz
IndexA2 = 1.649000	IndexB2 = 0.464000	IndexC2 = 0.129000
FreqA2 = 440 Hz	FreqB2 = 440 Hz	FreqC2 = 440 Hz



(a)

DFM OPERATOR A	DFM OPERATOR B	DFM OPERATOR C
AmpA = 0.546932	AmpB = 1.056423	AmpC = 3.433636
IndexA1 = 2.367000	IndexB1 = 4.377000	IndexC1 = 0.745000
FreqA1 = 220 Hz	FreqB1 = 220 Hz	FreqC1 = 440 Hz
IndexA2 = 1.607000	IndexB2 = 0.509000	IndexC2 = 0.189000
FreqA2 = 440 Hz	FreqB2 = 440 Hz	FreqC2 = 220 Hz



(b)

Fig. 8. DFM parameters of harpsichord. (a) Using genetic algorithm. (b) Using genetic annealing algorithm.

eters estimated by the genetic annealing algorithm are capable of generating spectra that are much closer to the real samples of the instruments from the MUMS CDs than by using the genetic algorithm alone.

**6 CONCLUSION**

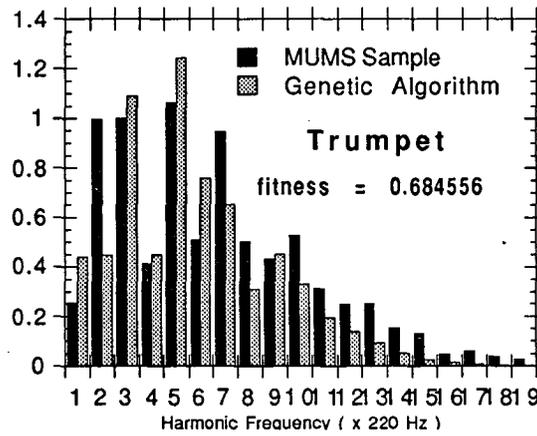
In this paper we have proposed a new combinatorial optimization technique which we call the genetic annealing algorithm (GAA). In this algorithm, simulated annealing, first proposed by Kirkpatrick et al. [5], is combined with the simple genetic algorithm, proposed by Holland [6], such that the crossover process of the genetic algorithm is modified to incorporate a simulated annealing-like procedure.

In physical terms we are assuming the fact that off-

spring often possess characteristics that resemble both parents. Thus in order to produce good offspring, we need two parents which possess different sets of good characteristics to cross in such a way that most of the good characteristics of each parent are inherited by the offspring and most of the bad ones are eliminated. This is done in our algorithm by making the best individual in the population (the one that contains the most good characteristics but is still lacking in some other good characteristics) in the current generation mate with several other individuals in the current generation. In this way, the best individual has a chance of meeting one or more partners that contain other required good characteristics which it is lacking.

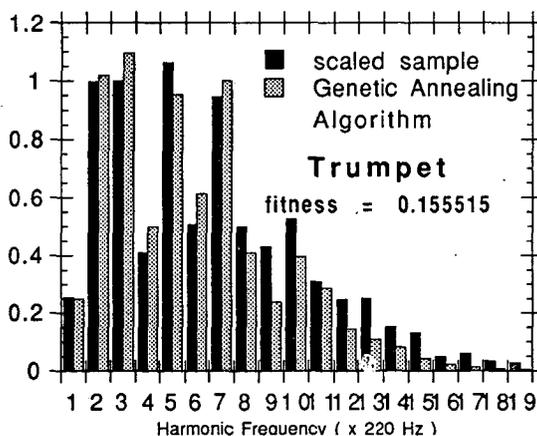
Furthermore, we need to cross the best individual with a partner that is at the most optimum bit position possible

DFM OPERATOR A	DFM OPERATOR B	DFM OPERATOR C
AmpA = 2.948724	AmpB = 3.374904	AmpC = 5.366608
IndexA1 = 2.503000	IndexB1 = 0.019000	IndexC1 = 1.535000
FreqA1 = 440 Hz	FreqB1 = 440 Hz	FreqC1 = 440 Hz
IndexA2 = 5.722000	IndexB2 = 5.238000	IndexC2 = 1.939695
FreqA2 = 220 Hz	FreqB2 = 220 Hz	FreqC2 = 220 Hz



(a)

DFM OPERATOR A	DFM OPERATOR B	DFM OPERATOR C
AmpA = 5.262182	AmpB = 2.815760	AmpC = 4.787540
IndexA1 = 1.390000	IndexB1 = 4.835000	IndexC1 = 5.246000
FreqA1 = 220 Hz	FreqB1 = 220 Hz	FreqC1 = 220 Hz
IndexA2 = 1.288000	IndexB2 = 3.326000	IndexC2 = 0.479000
FreqA2 = 440 Hz	FreqB2 = 440 Hz	FreqC2 = 440 Hz



(b)

Fig. 9. DFM parameters of trumpet. (a) Using genetic algorithm. (b) Using genetic annealing algorithm.

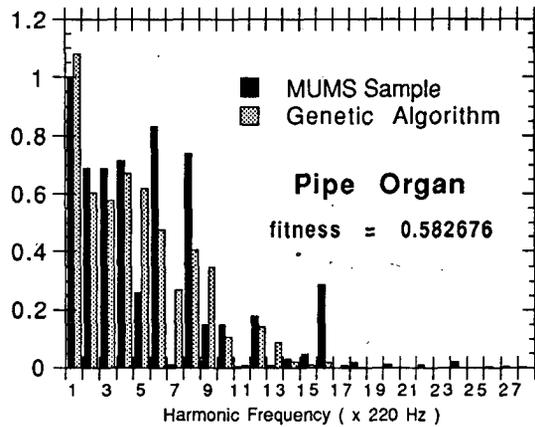
in order to generate offspring that are as improved as possible. To achieve this, each partner is allowed to generate offspring with the best individual several times. The improved offspring replace the parents, whereas the others may replace them after evaluation according to the Boltzmann distribution. The advantage of this algorithm over the simple genetic algorithm is that this algorithm will result in better optimized offspring in a smaller number of generations.

The best individual selected as the initial Parent\_A in genetic annealing is analogous to the initial state defined in simulated annealing. In standard simulated annealing the initial state is perturbed by randomly generating neighboring states whose energy levels are compared with the initial state. In genetic annealing we make use of the random crossover process of the genetic algorithm

to generate the random states with which to compare the initial state. The advantage of our algorithm over simulated annealing is that the random states (or offspring) generated are not random neighboring states but, being the result of crossovers within a population already selected for superior fitness, they have a better chance of being fitter than the parents.

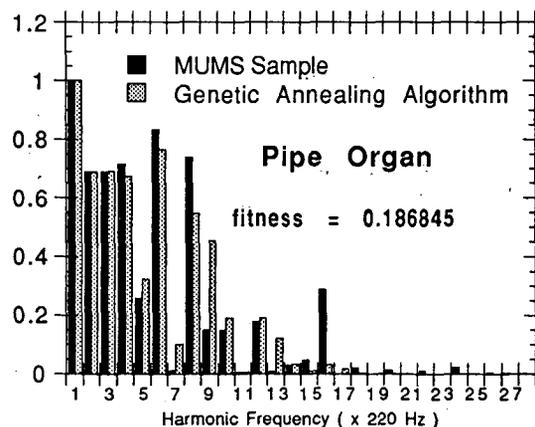
In Section 5 we found that the parameters obtained by the genetic annealing algorithm are significantly better than those from the genetic algorithm. As shown, the spectra generated are very closely matched to those of the real MUMS samples. Such close matches are very difficult but not impossible, using the small number of DFM operators that we have used, if the normal trial-and-error method of obtaining the DFM parameters is used. Even by using the genetic algorithm, this close

DFM OPERATOR A	DFM OPERATOR B	DFM OPERATOR C
AmpA = 1.584622	AmpB = 3.929953	AmpC = 2.219697
IndexA1 = 2.040000	IndexB1 = 0.443000	IndexC1 = 0.953917
FreqA1 = 220 Hz	FreqB1 = 440 Hz	FreqC1 = 220 Hz
IndexA2 = 4.606000	IndexB2 = 3.024000	IndexC2 = 2.014000
FreqA2 = 440 Hz	FreqB2 = 220 Hz	FreqC2 = 440 Hz



(a)

DFM OPERATOR A	DFM OPERATOR B	DFM OPERATOR C
AmpA = 1.653801	AmpB = 2.743604	AmpC = 4.337077
IndexA1 = 4.587000	IndexB1 = 0.145000	IndexC1 = 0.774000
FreqA1 = 440 Hz	FreqB1 = 220 Hz	FreqC1 = 440 Hz
IndexA2 = 2.814000	IndexB2 = 2.896000	IndexC2 = 2.922000
FreqA2 = 220 Hz	FreqB2 = 440 Hz	FreqC2 = 220 Hz



(b)

Fig. 10. DFM parameters of pipe organ. (a) Using genetic algorithm. (b) Using genetic annealing algorithm.

match might be achieved only if either the number of DFM operators or the population size is increased. This new optimization algorithm has thus helped to realize more fully the potential of the DFM synthesis technique.

In conclusion, the genetic annealing algorithm has the best of both worlds. It can converge solutions to the most optimized state, just like the simulated annealing algorithm, while allowing a fast convergence to that optimized state, as in a genetic algorithm.

## 7 REFERENCES

- [1] J. M. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *J. Audio Eng. Soc.*, vol. 21, pp. 526–534 (1973 Sept.).
- [2] B. T. G. Tan and S. L. Gan, "Real-Time Implementation of Asymmetrical Frequency-Modulation Synthesis," *J. Audio Eng. Soc. (Engineering Reports)*, vol. 41, pp. 357–363 (1993 May).
- [3] B. T. G. Tan, S. L. Gan, S. M. Lim, and S. H. Tang, "Real-Time Implementation of Double Frequency Modulation (DFM) Synthesis," *J. Audio Eng. Soc.*, vol. 42, pp. 918–926 (1994 Nov.).
- [4] B. Schottstede, "The Simulation of Natural Instrument Tones Using Frequency Modulation with a Complex Modulating Wave," *Computer Music J.*, vol. 1, no. 4, pp. 46–50 (1977).
- [5] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671–680 (1983 May).
- [6] J. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975).
- [7] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, MA, 1989).
- [8] A. Horner, J. Beauchamp, and L. Haken, "Machine Tongues XVI: Genetic Algorithms and Their Application to FM Matching Synthesis," *Computer Music J.*, vol. 17, pp. 17–29 (1993 Winter).
- [9] D. E. Brown, C. L. Huntley, and A. R. Spillane, "A Parallel Genetic Heuristic for the Quadratic Assignment Problem," in *Proc. 3rd Int. Conf. on Genetic Algorithms* (Arlington, VA, 1989, pp. 406–415).

## THE AUTHORS



B. T. G. Tan

B. T. G. Tan graduated in 1965 with a B.Sc. (Hons) in physics from the University of Singapore, and in 1968 with a Ph.D. from Oxford University. He is a chartered engineer and a member of the Institution of Electrical Engineers.

Since 1968 Dr. Tan has taught at the National University of Singapore where he is now an associate professor in physics. His research interests include the microwave



S. M. Lim

properties of semiconductors and dielectrics, surface physics, image processing, and digital sound synthesis.

•

S. M. Lim obtained a B.Sc. (Hons) in physics in 1994 from the National University of Singapore. She is currently working towards an M.Sc. in the field of digital sound synthesis at the National University of Singapore.