Performance of the Genetic Annealing Algorithm in DFM Synthesis of Dynamic Musical Sound Samples*

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

Department of Physics, National University of Singapore, Singapore 119260, Republic of Singapore

A qualitative analysis of the performance of a genetic annealing algorithm (GAA) is presented. The GAA is applied in the double frequency modulation DFM parameter optimization problem. It is shown that the GAA can optimize the solution to the global minimum by plotting the solution spaces of one DFM equation for all samples. Dynamic musical sounds were synthesized using the DFM synthesis technique. A recycling process is also proposed to improve the efficiency of the GAA process. The resulting dynamic sounds synthesized by the DFM are presented and compared with those of real samples.

0 INTRODUCTION

A variant of the frequency modulation (FM) technique called double frequency modulation (DFM) [1] technique has been shown to be able to generate more complex spectral envelopes than FM. One carrier of the DFM equation is given by

$$x(t) = A(t) \sin[I_1(t) \sin \omega_1 t + I_2(t) \sin \omega_2 t] \quad (1)$$

where $A(t)$ is the time-varying amplitude, $I_1(t)$ and $I_2(t)$ are the time-varying indices, and ω_1 and ω_2 are the corresponding modulating angular frequencies of the DFM carrier.

The behavior of the DFM equation is summarized in Fig. 1. Fig. 1(a) shows the variation of spectra in the frequency domain when index I_1 is varied whereas Fig. 1(b) shows the variation with index I_2, both keeping the modulation frequencies f_1 and f_2 fixed at 440 and 880 Hz. As shown in the plots, each harmonic oscillates in its own manner, independent of the other harmonics, as one index increases and the other is kept constant. We also note that as the value of either of the indices increases, the frequency of oscillation as well as the number of frequency harmonics increases.

Due to the complexity of the DFM equation, parameter estimation by the usual trial and error method is a very tedious process. Computer optimization is therefore essential if the process is to be speeded up. A combinatorial optimization technique, the genetic annealing algorithm (GAA), has been proposed in [2] and has been shown to be more effective than the classical genetic algorithm for DFM parameter optimization for steady-state musical sounds. It will be shown that the GAA is able to optimize the DFM parameters to the global minimum of the solution spaces for all steady-state sound samples in this paper.

However, to create convincing dynamic musical sounds, the harmonics of the sounds must vary over time. In this paper we show that the GAA can effectively estimate the DFM parameters quickly enough for practical determination of the time-varying harmonic structure of real musical sounds. We also propose a recycling process for the GAA to improve the efficiency of the optimization process.

1 THE OPTIMIZATION PROCESS

The data flow diagram for the optimization process is given in Fig. 2. The two-dimensional array created from the sampling process is used as input to the optimization algorithm. The algorithm used for dynamic or time-varying DFM parameter optimization is the genetic annealing algorithm (GAA) [2], a special combination of the simulated annealing algorithm and the genetic algorithm.

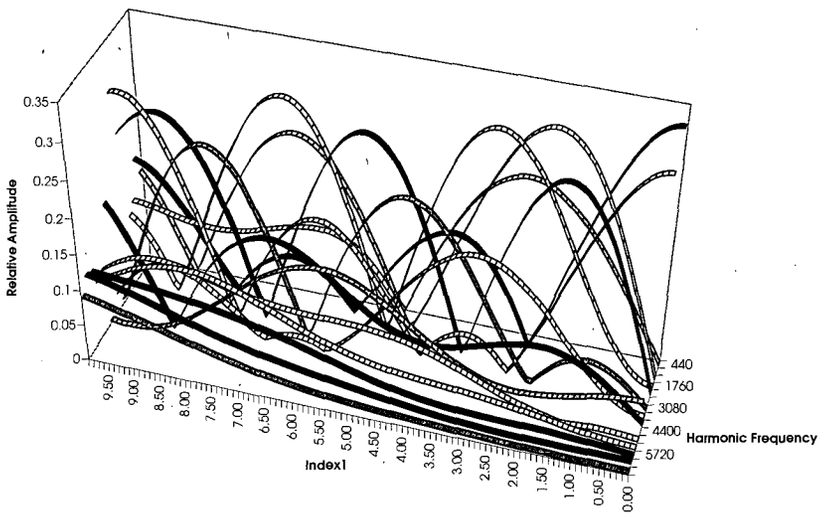
The simulated annealing algorithm and the genetic algorithm are both well-known algorithms in optimization theory and have been explained in depth by many authors. We will just briefly state the principles for the two algorithms here.

* Manuscript received 1998 March 14; revised 1999 February 2.

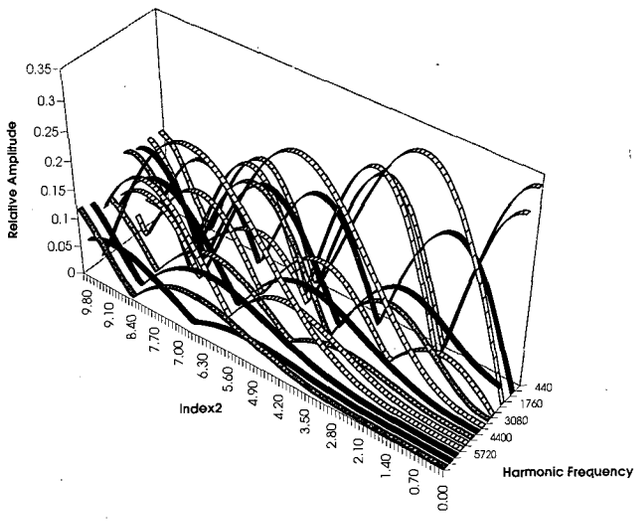
** Currently with the Centre for Signal Processing, Nanyang Technological University, Republic of Singapore.

1.1 Simulated Annealing Algorithm and Its Performance

The simulated annealing algorithm is analogous to the annealing process in solid-state physics. This algorithm was first proposed and applied by Kirkpatrick et al. [3] in combinatorial optimization problems. The simulated annealing process for a combinatorial optimization problem starts with an initial set of parameters at a predefined "temperature." This set of parameters is perturbed to a random neighboring state. If the fitness of this neighboring state is better than the initial state, it will be treated as the next initial state for the next perturbation. On the other hand, if the neighboring state is less fit than the initial one, it will only be accepted with a probability characterized by the Boltzmann distribution. Several random neighbors are generated until near "thermal equilibrium" is reached. Thermal equilibrium is said to be reached when there is no further decrease in the fitness value of the current state on further perturbations. The predefined "temperature" is then decreased representing cooling of the system, and the whole process repeats using the next initial state. The advantage of this algorithm is its ability to explore nearby regions of space to reach the minima of the solution spaces. In the process it is able to climb over small hills to reach nearby minima. But this kind of hill climbing search strategy often leads to solutions being trapped in regions of the solution space that have minima that are far from the optimal solution and eventually ending up at a local minimum. This search strategy thus only works well in a solution landscape like that shown in Fig. 3, as described in [4], which has low hills over which the algorithm can climb easily to seek the global minimum. Fig. 4 is not a good solution landscape for the simulated annealing algorithm since there are many minima enclosed by many tall barriers, which the algorithm is unable to overcome. The simulated annealing algorithm, however, will successfully reach the local minimum of the region where it first started. Hence a very good initial state is required for the simulated annealing algorithm to work in a landscape shown in Fig. 4, but this is often not the case.



(a)



(b)

Fig. 1. Variation of spectrum. (a) With DFM index1 when index2 = 3.0. (b) With DFM index2 when index1 = 3.0.

1.2 Genetic Algorithm and Its Performance

The genetic algorithm, first proposed by Holland [5], is based on the natural method by which genes combine and propagate in living organisms. A simple genetic algorithm consists of four main processes, namely, the recruitment, selection, crossover, and mutation processes.

The recruitment process randomly generates several sets of allowed values for the parameters of the problem to be optimized, with each set representing an individual, which will be encoded into bit strings. The collection of individuals makes up an initial population. Each individual is then subjected to a fitness test. The selection process selects the above average individuals and retains them in the population. The crossover process mates all the above average individuals present in the population to generate offspring that will make up the next generation by combining characteristics of the parents. In this way, above average sets of values will be inherited by the offspring. Occasional mutation is done on some offspring to simulate mutation of sites in the bit string (or chromosome).

The advantage of this algorithm is its ability to explore the whole region of the solution space to find a region in which the global minimum resides. Hence it can work better than the simulated annealing algorithm in a solution space whose landscape is as shown in Fig. 4. This is due to the fact that the genetic algorithm starts with a population that is well distributed over the whole solution space and has a high probability of being in or near the valley in which the global minimum lies. Moreover, since the valleys in Fig. 4 are very steep, if an initial state is in the current valley, it will be very close to the global minimum. The selection process ensures that the fitter individuals should have a higher chance of appearing in the population of the next generation after crossover. However, after several generations, the current best individual, which might not be the best global solution, often multiples to a stage whereby it dominates the population. This often results in loss of diversity, thus making the crossover process less effective. Genetic drift in the population might also result since a potentially good schema might be eliminated from the population [6]. Moreover, due to the random nature of the crossover process, there is a high probability that the resultant offspring is far away from the parents. Hence, upon reaching a nearby region in which the global minimum resides, the classical genetic algorithm may not be able to converge to that global minimum as proximity in the solution landscape does not confer additional advantages of being in the next generation, as is the case for simulated annealing. Hence the genetic algorithm might not be able to reach the global minimum in the solution landscape shown in Fig. 3.

The GAA, a combination of the genetic and the simulated annealing algorithms, has been proposed by Tan and Lim [2] and was proven more effective than the classical genetic algorithm for DFM synthesis [1] parameter optimization for steady-state musical sounds. This paper will provide a more detailed qualitative explanation of the behavior of the GAA by considering a general solution landscape.

To show that the GAA is able to optimize to the global minimum for our combinatorial problem, the DFM parameter optimization problem, the solution spaces of one DFM carrier for all our samples are plotted for comparison in Section 1.5. The next section explains how the GAA is superior to both the classical genetic algorithm and the simulated annealing optimization algorithm.

1.3 Theory of the Genetic Annealing Algorithm

The GAA is a genetic algorithm with a crossover process, which performs a simulated annealing-like procedure.

As the genetic algorithm often results in premature convergence of the solution due to loss of diversity and simulated annealing may end up in nonglobally optimal local minima due to the solution being constrained to only a local region of the solution space, a combination

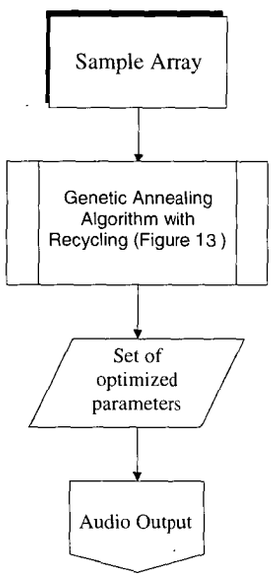


Fig. 2. GAA optimization process.

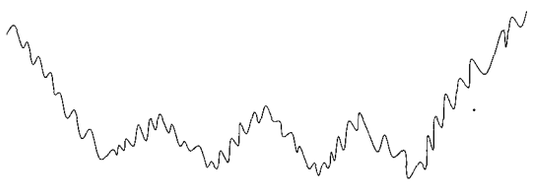


Fig. 3. Good solution space for simulated annealing algorithm.

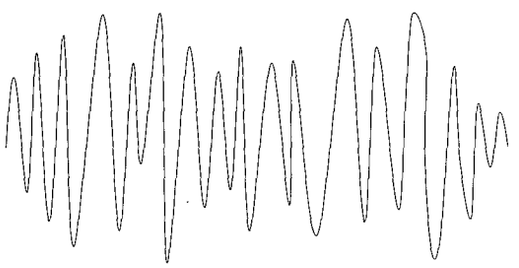


Fig. 4. Bad solution space for simulated annealing algorithm.

of the two algorithms, which we call the GAA, is able to combine the advantages of each algorithm. The GAA consists of a recruitment process followed by a selection process and ends with an anneal_cross process. This anneal_cross process is so named because it performs a simulated annealing procedure while doing the normal crossovers. The details of the methodology have been described in [2]. We will explain how the algorithm is able to work well on complex solution landscapes.

While the simulated annealing algorithm is efficient in solution spaces with landscapes of the type shown in Fig. 3 and the genetic algorithm works in solution spaces with landscapes shown in Fig. 4, the GAA is effective in both landscapes or a combination of the landscapes, as shown in Fig. 5.

The original GAA starts with a recruitment process, followed by a selection process, and ends with an anneal_cross process as described in [2]. In the recruitment process, initial random individuals are generated which are spread out rather evenly in the entire solution space. As in the genetic algorithm, the fitter individuals are selected for reproduction in the next generation in the selection process. From the fitter individuals, the best is selected and the anneal_cross process is performed on it. This simulated annealing-like process is done through crossing the fittest individual with several fit individuals, in the process generating new offspring.

Unlike the normal crossover process, only those individuals better than the direct parents are retained in the population. Those that are less fit than the parents will be retained only with a probability given by the Boltzmann distribution. This is to enable small barriers to be overcome to seek another region with a more optimum minimum. This follows the procedure in simulated annealing.

The nature of our crossover process (Fig. 6) shows that the offspring may be either near to or far from its direct parents. For the example shown, the values of the two parents (string1 and string2) are 6.053 and 2.138. These two values yield a possible range of values, as

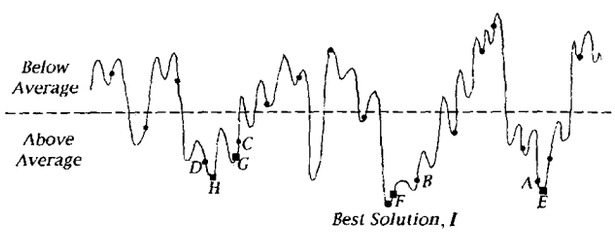


Fig. 5. Example solution space for genetic annealing algorithm.

	Binary Strings																Decimal
Bitnumber	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Value
Parents:																	
String1	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	6.053
String2	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	0	2.138
Offsprings obtained :																	
after Crossing at Bit0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	0	6.052
	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	2.139
after Crossing at Bit1	0	0	0	1	0	1	1	1	1	0	1	0	0	1	1	0	6.054
	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1	2.137
after Crossing at Bit2	0	0	0	1	0	1	1	1	1	0	1	0	0	0	1	0	6.05
	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	2.141
after Crossing at Bit3	0	0	0	1	0	1	1	1	1	0	1	0	1	0	1	0	6.058
	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	1	2.133
after Crossing at Bit4	0	0	0	1	0	1	1	1	1	0	1	1	1	0	1	0	6.074
	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	2.117
after Crossing at Bit5	0	0	0	1	0	1	1	1	1	0	0	1	1	0	1	0	6.042
	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	1	2.149
after Crossing at Bit6	0	0	0	1	0	1	1	1	1	1	0	1	1	0	1	0	6.106
	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	2.085
after Crossing at Bit7	0	0	0	1	0	1	1	1	0	1	0	1	1	0	1	0	5.978
	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	1	2.213
after Crossing at Bit8	0	0	0	1	0	1	1	0	0	1	0	1	1	0	1	0	5.722
	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1	2.469
after Crossing at Bit9	0	0	0	1	0	1	0	0	0	1	0	1	1	0	1	0	5.21
	0	0	0	0	1	0	1	1	1	0	1	0	0	1	0	1	2.981
after Crossing at Bit10	0	0	0	1	0	0	0	0	0	1	0	1	1	0	1	0	4.186
	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1	4.005
after Crossing at Bit11	0	0	0	1	1	0	0	0	0	1	0	1	1	0	1	0	6.234
	0	0	0	0	0	1	1	1	1	0	1	0	0	1	0	1	1.957
after Crossing at Bit12	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	0	2.138
	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	6.053
after Crossing at Bit13	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	0	2.138
	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	6.053
after Crossing at Bit14	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	0	2.138
	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	6.053
after Crossing at Bit15	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	0	2.138
	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	6.053

Fig. 6. Results of crossover of binary bit strings.

shown in Fig. 6. Crossing at bit 4 will yield offspring of 6.074 and 2.117 while crossing at bit 5, for example, yields 6.042 and 2.149, respectively, which are very close to their parents and thus remain in the local region. We note that the offspring may thus position themselves on either side of their parents. This process therefore is equivalent to "small perturbation" in simulated annealing. For other values of the crossover bit, such as 8, 9, or 10, the offspring have values quite different from the parents. For example, the crossover bit at 10 yields offspring of 4.186 and 4.005, which are both far from the parents and most probably lying in other valleys.

The former case therefore descends the local valleys for better solutions of the simulated annealing kind, whereas the latter case explores other regions of the solution space for possible superior solutions as for the genetic algorithm. Both processes therefore contribute to finding a better solution. In parallel to this, the other offspring generated directly by the randomly chosen partners also perform either exploitation of the local region or exploration of remote regions in which a better solution may lie.

Hence for every crossover process, the chances are that either two valleys are descended, two new regions are explored, or one valley is descended and one new region explored. No matter which case occurs, it is likely to contribute to an improved solution in a manner superior to either the genetic algorithm or the simulated annealing algorithm alone.

An example solution space is plotted in Fig. 5. This solution space has multiple deep valleys and high hills, and within each there are further smaller hills and valleys. This complex solution space is a combination of Figs. 3 and 4. Take, for example, the small dots shown, which are individuals in the solution space generated as the initial population. The best individual thus far is state A. This individual is crossed with several above average individuals to generate offspring. In cases when its offspring are neighboring, state A will try to exploit its own valley and finally approaches point E (the local minimum). At the same time, its other partners, say, C and D, will also be perturbed and be shifted to their neighboring states, say, G and H, respectively, their fitness calculated and compared with A. In cases where the offspring are much further from the parents and in different valleys, the fitness of these new offspring is also calculated to find possible valleys containing a better solution. The Boltzmann distribution will allow state B to cross also the small barrier to the left and proceed to state F, which is fitter than the optimum state E first generated by state A. At this point, the fittest solution chosen for the next anneal_cross process will be state F. This current best state F will then undergo a simulated annealing-like process by generating its neighboring states and finally reaching the best solution, I, which is the global minimum.

The random selection process has thus enabled the whole solution space to be explored for a set of above average solutions. The anneal_cross process on the other hand helps exploit the nearby states of the set of better solutions and in the process is also able to help them climb over small barriers, so as to reach the local minimum of each of the regions. Since the anneal_cross process functions by a normal crossover process, it is also able to explore other further regions of space for other possible solutions.

It should be noted that the selection process to select the above average solutions for further exploitation is only done once in GAA. In this way, the possibility of loss of diversity will not result since the population will always contain both the currently very fit individuals as well as the less fit ones. In fact, all the selected individuals will be given a chance to improve to their maximum potentials since none will be eliminated in a GAA process. The problem of genetic drift is thus eliminated.

In general, all individuals will seek to attain their own local minima in parallel. That is, as shown in Fig. 5, D attains its local minimum at H whereas C attains its local minimum at G, A attains its local minimum at E, and B seeks the best solution after crossing a barrier, all in parallel. However, offspring which are in different valleys from their parents, but fitter, will be retained and thus enable new and superior regions not occupied by the original set of selected individuals to be explored. Hence with the GAA, there is a high probability of obtaining a solution at the global minimum.

1.4 Implementation of the GAA in DFM Optimization

The GAA was implemented using the C language programming. The pseudo code is shown in Fig. 7. The GAA consists of three major processes, namely, the recruitment process, the selection process, and the anneal_cross process.

Recruitment Process. The recruitment process starts with the generation of new random individuals to create the initial population. It uses the uniform random number generator to generate the two indices of the DFM parameters and the two numbers representing the harmonic numbers of the two modulating frequencies. For our application, the values of the initial index range between zero and the MaxIndex, a control parameter defined by the user which is normally set to be about 10, for our case. On the other hand, the frequencies can take a value of any of the harmonic frequencies of the sample. Using the set of generated random parameters, the fitness value of each individual is calculated by taking the total of the squares of the normalized differences between the harmonic amplitudes of the synthesized and the sample spectra. The lower this value, the fitter the individual will be.

Selection Process. The selection process we employ is the simple binary-tournament-selection [7] technique. All selections are based on the fitness value of each individual. Hence after the selection process above average individuals will show up in the next generation.

Anneal Cross Process. This is the process that determines the performance of the algorithm. The process starts with choosing the best individual, based on its fitness value, as one of the parents. This best individual

is then allowed to undergo several crossover processes with each of the other randomly selected individuals from the current population. The number of individuals selected for mating is defined to be proportional to the number of harmonics in the sample. The number of crossovers with the Kth individual selected is given by $P*K/H*0.2$, where P is the population size and H is the total number of harmonics in the sample. The initial fitness temperature is defined to be equal to a value equal to about 10% of the total harmonic amplitudes of the sample spectrum.

Each mating process performs a parameter crossover on either index1, index2, modulating frequency 1, or modulating frequency 2 between the two parents. After each crossover process, the fitness of each of the two offspring is evaluated and compared with their direct parents. If the fitness is improved, it will replace its direct parent in the current population. Otherwise it will replace the parent with an acceptance probability characterized by the Boltzmann distribution,

$$P(\text{Boltzmann}) = \exp\left(\frac{\text{newfit} - \text{oldfit}}{\text{fitness Temp}}\right).$$

This acceptance probability is computed and compared with a random number generated from the uniform random number generator. If this random number is less than the Boltzmann probability calculated, the offspring would be accepted or else discarded.

1.5 Results

We have used the genetic annealing algorithm and the genetic algorithm to optimize the DFM parameters for synthesizing samples of violin, saxophone, piano, oboe, and trumpet sounds at the pitch of A4. The samples were obtained from McGill University master sample CDs, and the source information is given in Table 1.

To show that our GAA optimizes to the best parameters, the one-carrier DFM solution space for randomly selected time segments of each sample was plotted (see Figs. 8–12), varying only the indices while keeping f_1 and f_2 fixed at 440 and 880 Hz, respectively. The z axis

Table 1. Sample sources from McGill University master CD.

Instrument	Figure	Volume	Track	Index
Violin	8	1	1	16
Saxophone	9	3	16	09
Piano	10	3	3	16
Oboe	11	11	81	11
Trumpet	12	8	57	02

```
Begin
    Get_Sample();
    Recruit();
    Select();
    DO
      Anneal_Cross();
    WHILE (Bestfitness is not satisfactory or
           Time's Up);
End
```

```
Anneal_Cross()
ParentA = ChooseBest()
FitnessTemperature = 0.1
FOR K = 1 TO No_Of_Harmonics DO
    Choose the Kth Partner
    ParentB = ChoosePartner()
    FitnessTemperature = FitnessTemperature * 0.9
    FOR Tk = 1 TO (Npop * K/ No_Of_Harmonics * 0.2) DO
        Generate Tk Offspring
        Crossbits(Random_Bit_Number)
        If Fitness(OffspringA) is better than Fitness(ParentA)
            Update(ParentA <= OffspringA)
        ElseIf RandomNum(1) < Boltz_Prob(Fitness(OffspringA)
            BackUpBest(ParentA)
            Update(ParentA <= OffspringA)
        If Fitness(OffspringB) is better than Fitness(ParentB)
            Update(ParentB <= OffspringB)
        ElseIf RandomNum(1) < Boltz_Prob(Fitness(OffspringB)
            Update(ParentB <= OffspringB)
    END FOR
END FOR
```

Fig. 7. Genetic annealing algorithm pseudo code.

gives the fitness of the corresponding DFM equation on a logarithmic scale whereas the x and y axes correspond to indices I_1 and I_2 with steps of 0.05. The global minimum of the fitness value was determined and compared with the optimum minimum obtained by the GAA. It was found that for all our samples, despite the irregular and unpredictable terrain in the solution spaces as shown, the optimized DFM indices generated from the GAA matched almost exactly the global minima of each solution space. For comparison, the classical genetic algorithm (GA) was also used for the optimization of the DFM parameters, and the best indices found using the GA are tabulated in Table 2 together with the global minima found by the GAA for the solution spaces. It was found that the classical GA is unable to optimize the global minimum of the solution spaces for most of our samples. The GAA was always found to be able to optimize the DFM parameters to a better fitness than the GA, which was very close to the actual global minimum. The results for each instrumental sound are given in Sections 1.5.1–1.5.5.

1.5.1 The Violin

Solution Space. The solution space for the spectrum of the violin at 250 ms is shown in Fig. 8. With f_1 and f_2 fixed at 440 and 880 Hz, respectively, the minimum fitness of 0.270 occurs at $I_1 = 1.45$ and $I_2 = 0.65$.

Performance of the GAA. Using the classical GA for one DFM carrier, the best indices found were $I_1 = 0.88$ and $I_2 = 0.46$ at a fitness value of 0.306. Using the GAA, the optimized parameters were found to be $I_1 = 1.46$ and $I_2 = 0.66$ at a fitness value of 0.260, which is at the global minimum. Hence the GAA was found to be much more effective in optimizing the DFM parameters for this spectrum to the global minimum compared to the GA.

1.5.2 The Saxophone

Solution Space. The solution space for the spectrum of the saxophone at 125 ms is shown in Fig. 9. With f_1 and f_2 fixed at 440 and 880 Hz, respectively, the minimum fitness occurs at $I_1 = 4.95$ and $I_2 = 1.35$ with a value of 0.530.

Performance of the GAA. Using the classical GA for one DFM carrier, the best indices found were $I_1 = 5.01$ and $I_2 = 1.45$. Using the GAA, the optimized parameters were found to be $I_1 = 4.98$ and $I_2 = 1.37$, which is very close to the global minimum, comparing with 4.95 and 1.35, respectively. The fitness values found were 0.557 and 0.533 for the parameters found by the GA and the GAA, respectively. The GAA optimization was again found to be more effective compared to the GA for this spectrum.

1.5.3 The Piano

Solution Space. The solution space for the spectrum of the piano at 2125 ms is shown in Fig. 10. With f_1 and f_2 fixed at 440 and 880 Hz, respectively, the minimum fitness of the graph occurs at $I_1 = 0.65$ and $I_2 = 0.40$.

Performance of the GAA. Using the classical GA for one DFM carrier, the best indices found were $I_1 = 1.03$ and $I_2 = 0.53$. Using the GAA, the optimized parameters were found to be $I_1 = 0.60$ and $I_2 = 0.39$. From Table 2, the fitness values were, respectively, 0.050 and 0.011 for the parameters found by the GA and the GAA. The GAA was also found to optimize the DFM parameters for this spectrum closer to the global minimum compared to using the GA.

1.5.4 The Oboe

Solution Space. The solution space for the spectrum of the oboe at 1500 ms is shown in Fig. 11. With f_1 and f_2 fixed at 440 and 880 Hz, respectively, the minimum of the graph occurs at $I_1 = 3.50$ and $I_2 = 1.50$.

Performance of the GAA. Using the classical GA for one DFM carrier, the best indices found were $I_1 = 3.47$ and $I_2 = 1.41$. Using the GAA, the optimized

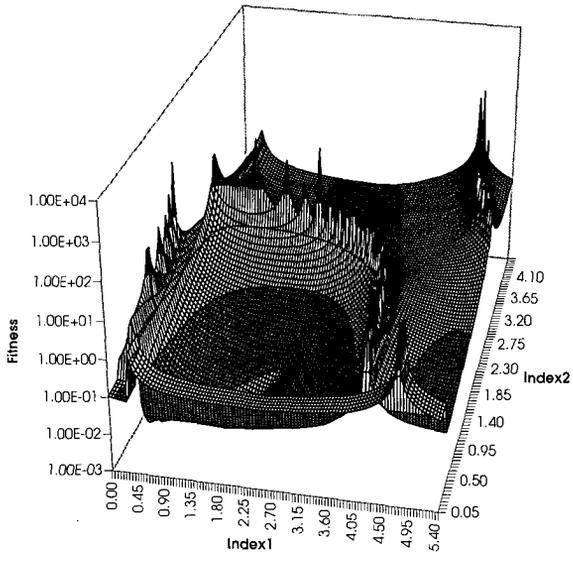


Fig. 8. Solution space for violin at 250 ms.

Table 2. Comparison of best indices found from solution space with best indices found from GAA and GA.*

	Minimum of Solution Space		GAA			GA		
Acoustic Instrument	I_1	I_2	Fitness	I_1	I_2	Fitness	I_1	I_2
Violin at 250 ms	1.45	0.65	0.260	1.46	0.66	0.306	0.88	0.46
Saxophone at 125 ms	4.95	1.35	0.533	4.99	1.37	0.557	5.01	1.45
Piano at 2125 ms	0.65	0.40	0.011	0.60	0.39	0.050	1.03	0.53
Oboe at 1500 ms	3.50	1.50	0.406	3.40	1.54	0.457	3.57	1.34
Trumpet at 2000 ms	3.05	1.40	0.312	3.07	1.38	0.331	2.22	1.82

* One operator of DFM is used, varying indices, keeping frequencies fixed at $f_1 = 440$ Hz and $f_2 = 880$ Hz.

parameters were found to be $I_1 = 3.39$ and $I_2 = 1.47$. Comparing with 3.50 and 1.50, respectively, we found that both the GAA and the GA could optimize the DFM parameters for this spectrum rather close to the global minimum. From Table 2, the solution optimized by the GAA has a fitness of 0.406 whereas that by the GA has a fitness value of 0.457, which indicates that the solution found by the GAA is a better one.

1.5.5 Trumpet

Solution Space. The solution space, with f_1 and f_2 at 440 and 880 Hz, respectively, for the spectrum of the trumpet at 2000 ms is shown in Fig. 12(a). Viewing the solution space of the trumpet from the sides yields Fig. 12(b) and (c). By careful observation of these solution spaces we found that there exist two major minima, one occurring at $I_1 = 3.05$ and $I_2 = 1.40$, the other at $I_1 = 2.25$ and $I_2 = 1.80$. Among these two minima, the former gives a lower (that is, better) fitness value of 0.312. Hence we may say that the global minimum is at $I_1 = 3.05$ and $I_2 = 1.40$.

Performance of the GAA. Using the classical GA for optimizing one DFM carrier, the best indices found were $I_1 = 2.22$ and $I_2 = 1.82$. Using the GAA, the optimized parameters were found to be $I_1 = 3.07$ and $I_2 = 1.38$. The fitness values found were 0.331 and 0.312 for the parameters found by the GA and the GAA, respectively. We found that the GA converges to its solution prematurely before the global minimum is reached. This confirms that the GAA is capable of optimizing the solution to the global minimum for this spectrum.

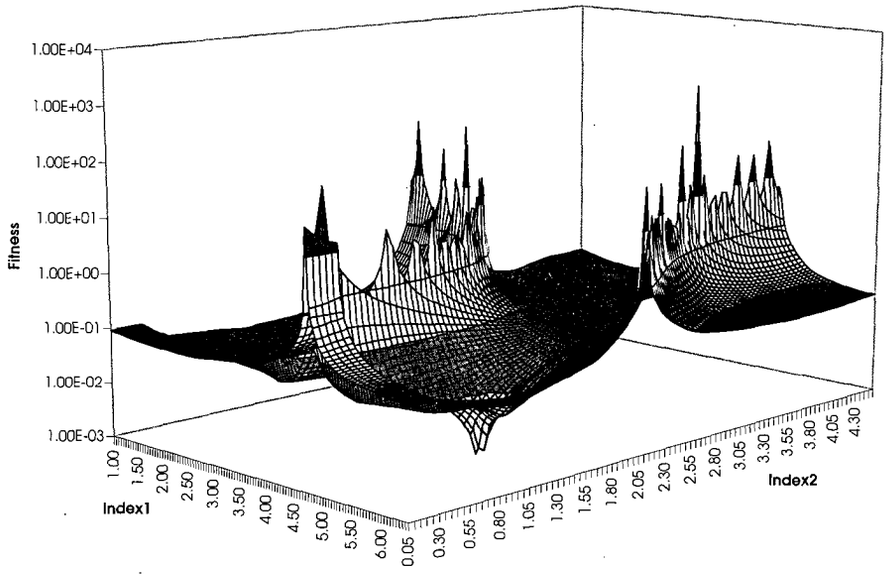


Fig. 9. Solution space for saxophone at 125 ms.

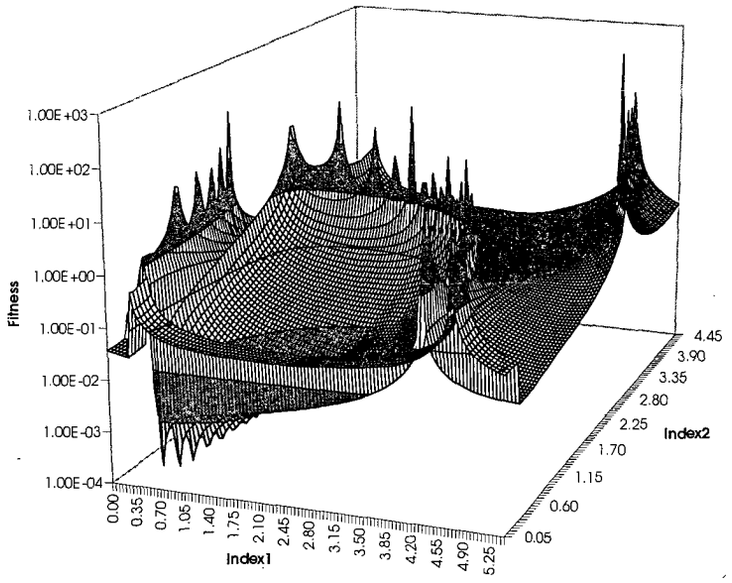


Fig. 10. Solution space for piano at 2125 ms.

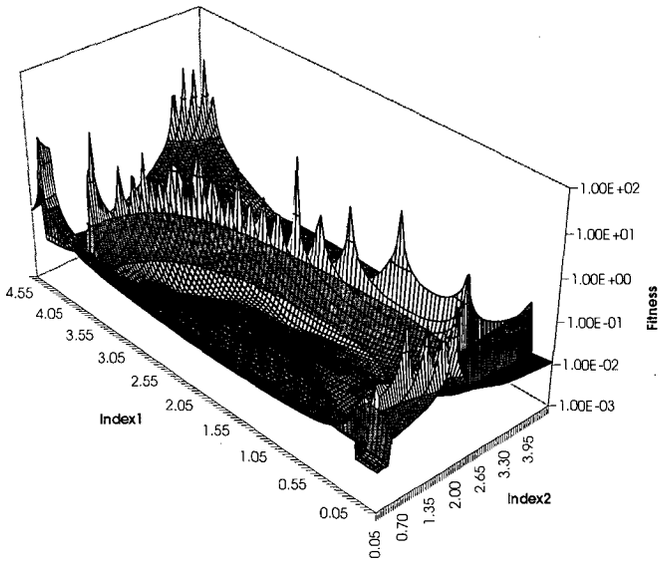
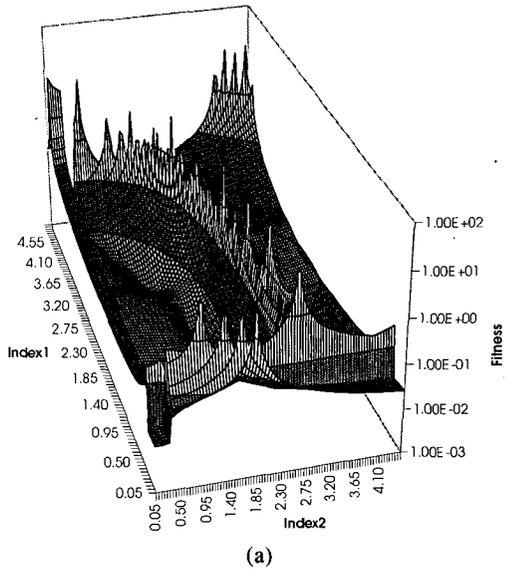
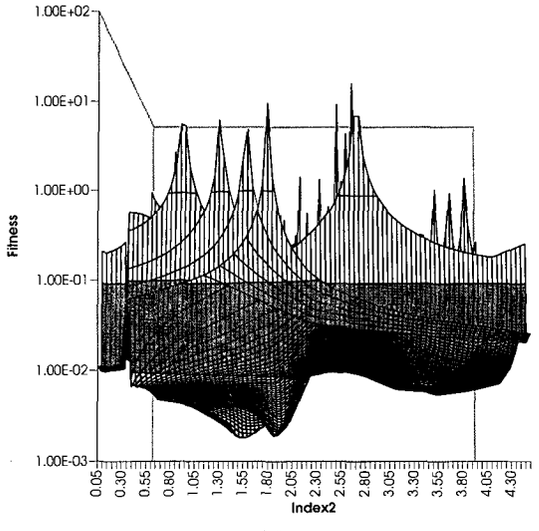


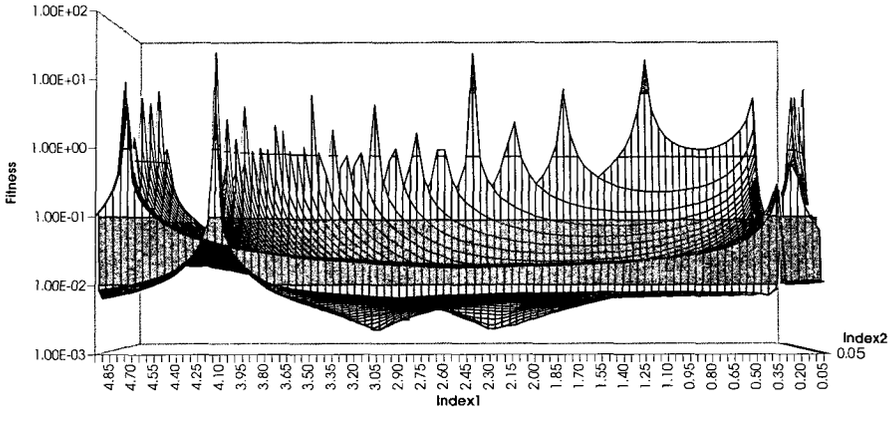
Fig. 11. Solution space for oboe at 1500 ms.



(a)



(c)



(b)

Fig. 12. Solution spaces at 2000 ms. (a) Trumpet. (b) Trumpet viewed from index1 axis. (c) Trumpet viewed from index2 axis.

2 DISCUSSION

In the previous sections we reviewed the GAA to explain how it is more effective in seeking the global minimum in the solution spaces for the DFM parameter optimization than both the simple genetic algorithm and the simulated annealing algorithm used separately. While the genetic algorithm works on the assumption that good schemas are derived only from good parents, the GAA assumes that each and every individual has some good schemas and has the potential to contribute to the best solution. Thus once they are selected by a preliminary selection process, none of them is eliminated throughout the GAA. Most individuals, while helping the best to improve further, undergo similar training processes by several interactions (or crossovers) with the best individual. In the process, all individuals have an equal chance to contribute to the best individual in the next generation since no further selection or elimination will take place. In this way no loss of diversity will occur as the population will contain individuals with diverse characteristics. Our results show that the algorithm is able of optimizing the global minima, as expected, for all our samples.

Three-dimensional solution spaces of selected time segments of all our samples were plotted. The solution spaces for one DFM carrier for all our samples resemble that shown in Fig. 5, which is a combination of the landscapes shown in Figs. 3 and 4. As shown in the results, all the GAA-optimized DFM parameters matched the global minima of the solution spaces of the respective samples while the classical genetic algorithm usually converges before the global minimum is reached for most of our samples (as shown in Table 2). This further confirms that the GAA is superior to the genetic algorithm in converging solutions to the global minima in complex and irregular solution landscapes. We therefore believe that the GAA can be used to solve many other combinatorial optimization problems with efficiency and accuracy.

3 SYNTHESIS OF DYNAMIC MUSICAL SOUNDS

Since for most musical notes, except at the attack and decay portions, the sound varies rather continuously, the optimized DFM parameters for two spectra of adjacent time frames will not vary too abruptly. For efficiency, we propose a recycling process just before the commencement of the calculation for the next time frame of the musical sound. This process is included as part of the recruitment process to ensure that the best solution obtained in the previous time frame appears as an individual in the initial population for the next. In other words, the recycling process helps to create a near optimized initial state for further optimization if the spectrum does not vary much compared to that of the previous time frame. This saves greatly on the time to reach the most optimized state since the algorithm does not have to start from scratch again with an initial random population at every time frame of the musical sample. This is especially important for the GAA as simulated annealing is the basis of its anneal–cross process. This process will perform very much more effectively if the initial state is near optimized. Even if the spectrum of the next time frame differs greatly from that of the previous time frame, the recycling process will not significantly degrade the performance of the algorithm. In this case, the recycled best individual from the previous time frame will act as one of the randomly generated individuals in the next recruitment process, and thus the whole GAA process will still perform efficiently.

The modified GAA with the recycling process for dynamic DFM optimization is clearly shown in Fig. 13. The whole process starts with the input of the frequency spectrum of the zeroth time frame of the real sample. This obviously does not require the recycling process and enters the GAA process directly for optimization of the DFM parameters using an initial random population sample. The optimized DFM parameters that are output from the GAA process are stored in an output file. The

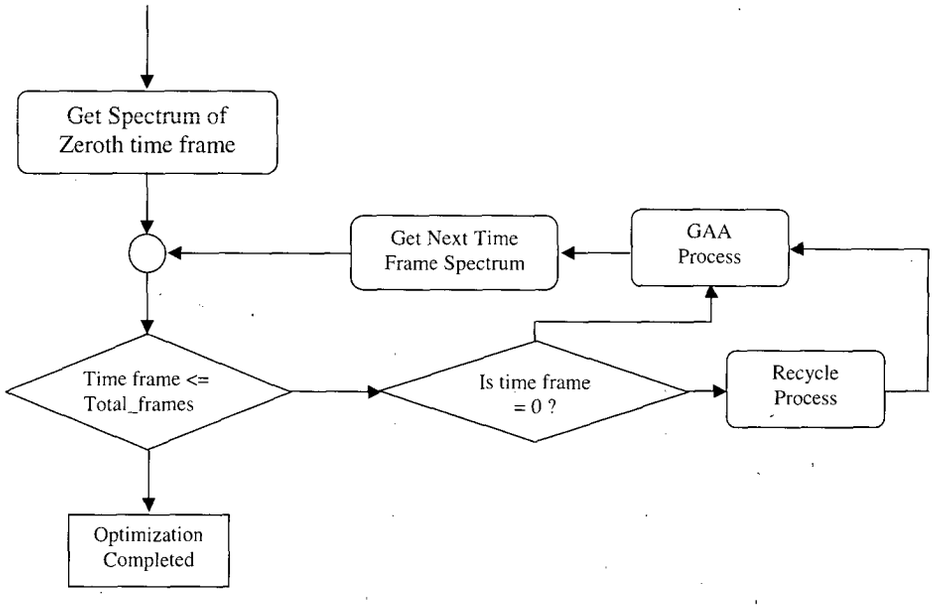


Fig. 13. Incorporating recycling process in GAA optimization.

frequency spectrum of the next time frame is then obtained and checked whether it is null, that is, the end of the real musical sound sample has been reached, which will terminate the process. Otherwise it will undergo the recycling process. The recycling process will capture the best set of parameters from the previous time frame and send it to the next recruitment process so that the previous best individual can be made one of the individuals in the next time frame. This whole process is repeated for all time frames before the whole optimization is completed. As will be expected, as the time frame number increases, the number of generations required to reach the optimized solution will decrease, especially for the sustain portion of a musical sound. The complete flow diagrams of our optimization process are given in Figs. 13 and 14.

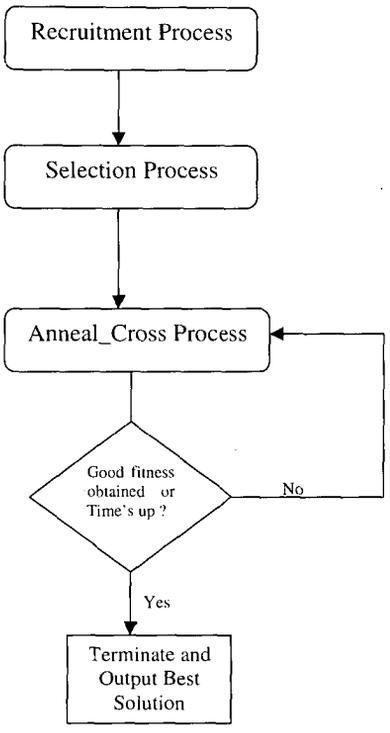


Fig. 14. Simple GAA process.

3.1 Results

The results of optimization for the time-varying harmonics of the samples are presented in the following sections. It was found that for all our samples, the time-varying harmonics could be synthesized to an acceptable degree of accuracy with the synthesized tones that resemble the sound of the real instruments, using only two DFM carriers. Note that a fitness value will be found for every time frame. To compare the results of one instrument with another, an average fitness is calculated by taking the mean of the sum of fitness values for all time frames (typically 150 frames) for each instrument tone. All the fitness values given in this section are the average fitness for that instrument (Fig. 15 and Table 3).

3.1.1 The Violin

The spectrum for the real violin sample is plotted in Fig. 16(a). As shown in the graph, the attack lasts for about 750 ms and sustains with a continually vibrating tone for 2500 ms. The resultant synthesized spectrum is shown in Fig. 16(b). The relative amplitudes of the resultant harmonics characterize those of the violin very closely with a fitness of 0.0260. For comparison, the fitness obtained using the classical GA with the recycling process was 0.0537, which was higher than that obtained by the GAA.

3.1.2 The Saxophone

The spectrum for the real saxophone sample is plotted in Fig. 17(a). Using the GAA the fitness obtained is 0.0149, which is lower than that obtained by the classi-

Table 3. Comparison of fitness values obtained from GAA and GA.

Optimization Method	GAA	GA
Violin	0.0260	0.0537
Saxophone	0.0149	0.0210
Piano	0.0234	0.0320
Oboe	0.0283	0.0616
Trumpet	0.0139	0.0148

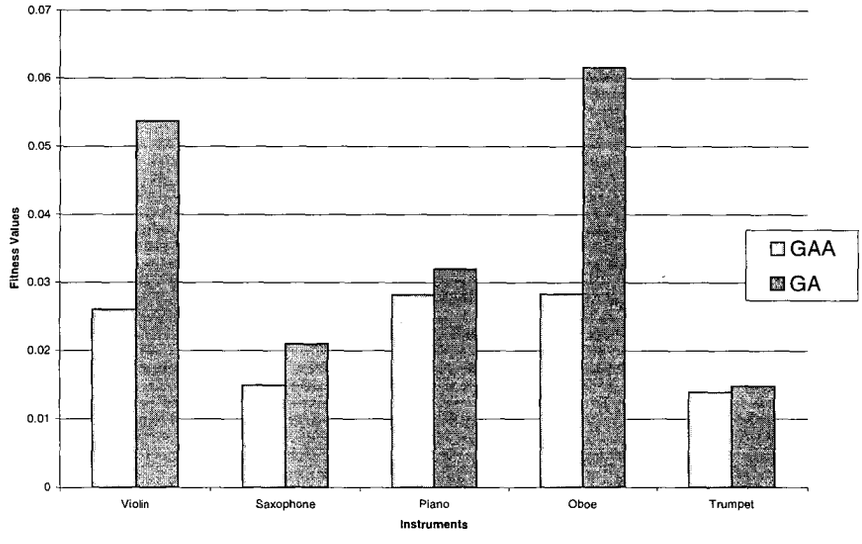


Fig. 15. Plot of fitness values obtained from GAA and GA. Lower fitness indicates better fit.

cal GA, namely, 0.0210. The relative amplitudes of the resultant harmonics and the envelopes optimized by the GAA in general characterize those of the saxophone. The entire synthesized spectrum for the synthesized saxophone is shown in Fig. 17(b).

3.1.3 The Piano

The spectrum for the real piano sample is plotted in Fig. 18(a). The relative amplitudes of the resultant harmonics characterize those of the piano very closely with a fitness of 0.0234. The resultant synthesized spectrum is shown in Fig. 18(b). The fitness obtained using the classical GA is found to be 0.0320, which is less optimum than that obtained from the GAA.

3.1.4 The Oboe

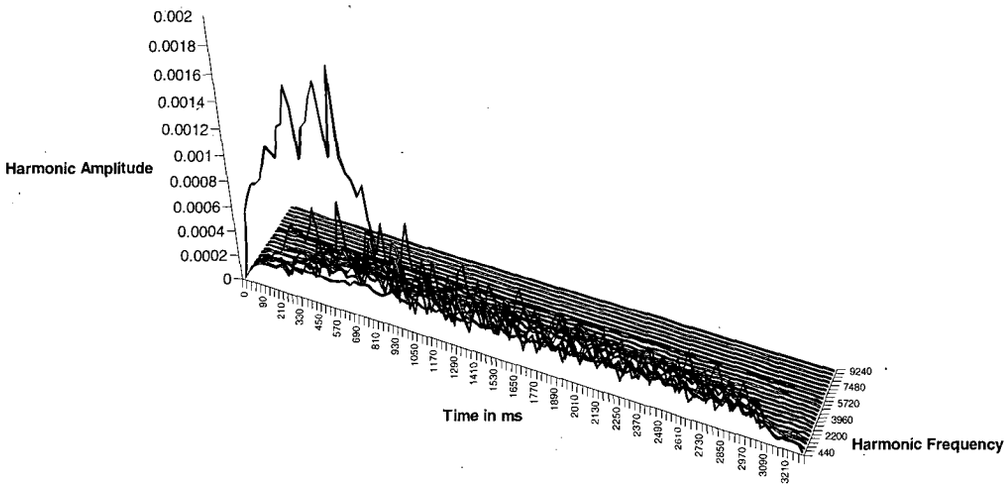
The spectrum for the real oboe sample is plotted in Fig. 19(a). Using the GAA the relative amplitudes of the resultant harmonics in general characterize those of the oboe with a fitness of 0.0283. The entire synthesized spectrum of the oboe is shown in Fig. 19(b). It is found that the synthesized one closely resembles that of the real oboe. Using the classical GA, the fitness obtained is 0.0616, which is less optimum.

3.1.5 The Trumpet

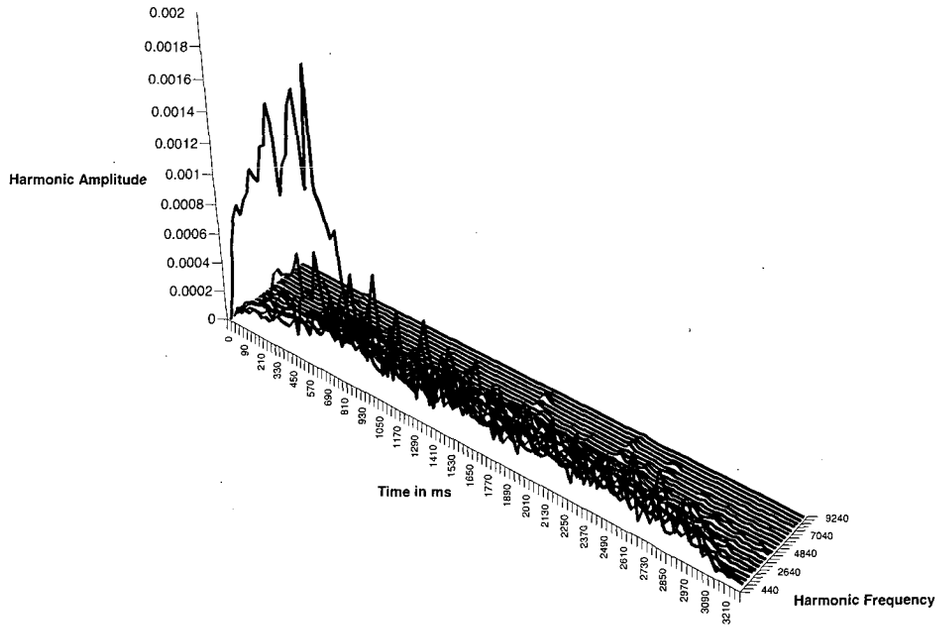
The spectrum for the real trumpet sample is plotted in Fig. 20(a). Using the GAA the fitness found is 0.0139, which is again more optimum compared with the fitness of 0.0148 obtained by the classical GA, and the spectrum of the synthesized trumpet by the former is shown in Fig. 20(b). The relative amplitudes of the resultant harmonics in general characterize those of the trumpet.

4 CONCLUSIONS

From the results we have shown that the GAA is effective in optimizing the DFM parameters for continu-



(a)



(b)

Fig. 16. Spectrum plots. (a) Sample violin at 440 Hz. (b) Synthesized violin at 440 Hz. Average fitness = 0.0260.

ous sounds. By incorporating a recycling process before the recruitment process of the GAA the DFM parameters could be optimized faster. This is because the initial state of the next time frame is usually near optimized since the time variation of the harmonics of real musical sounds is almost always smoothly continuous.

Our results also show that the DFM synthesis technique is capable of synthesizing the time-varying spectrum of musical sounds in general by using only two DFM carriers.

As our GAA technique of DFM parameter estimation is quite fast, with typical values of less than 7 s to achieve optimization for the first time frame of the spectrum of real music and about less than 3 s for latter time frames with the recycling process, this technique could be developed as a means of compressing real music by transmitting and storing only the DFM parameters of the time-varying sounds, which could then be reproduced by a DFM synthesis process in real time.

5 REFERENCES

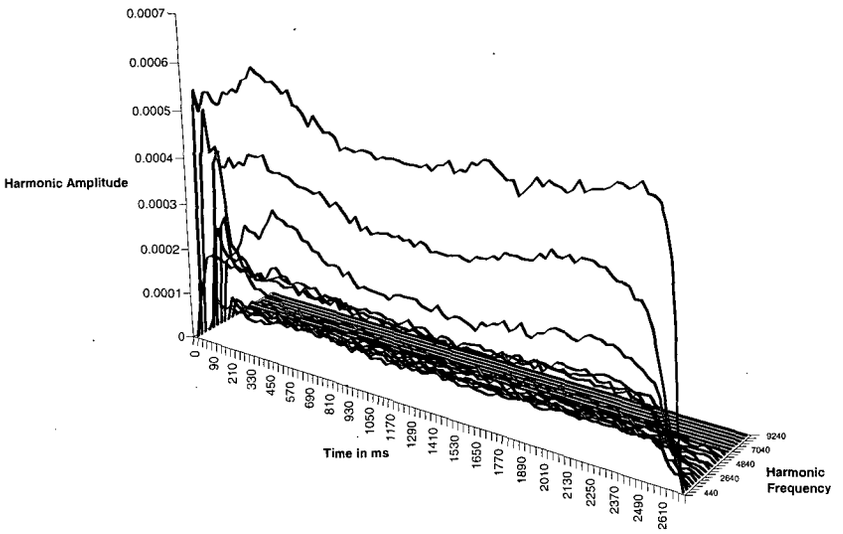
[1] B. T. G. Tan, S. L. Gan, S. M. Lim, and S. H. Tang, "Real-Time Implementation of Double Frequency Modulation (DFM) Synthesis," *J. Audio Eng. Soc.*, vol. 42, pp. 918–926 (1994 Nov.).

[2] B. T. G. Tan and S. M. Lim, "Automated Parameter Optimization for Double Frequency Modulation Synthesis Using the Genetic Annealing Algorithm," *J. Audio Eng. Soc.*, vol. 44, pp. 3–15 (1996 Jan./Feb.).

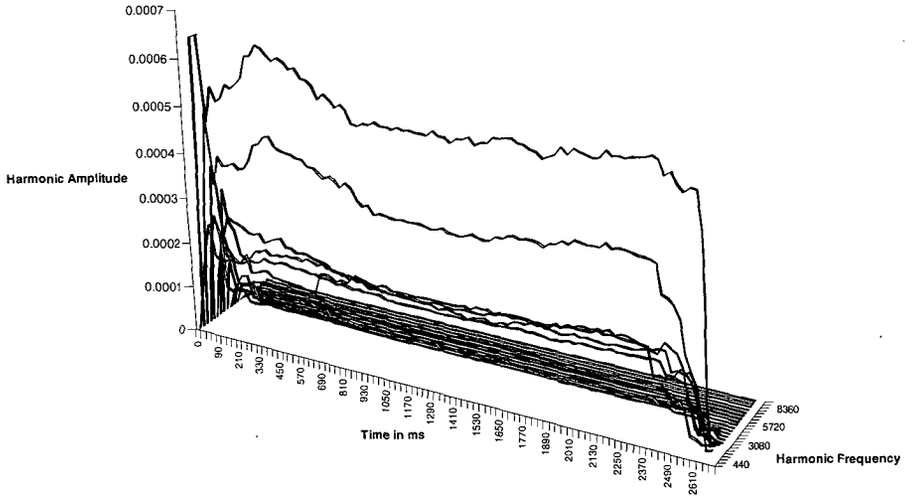
[3] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vechi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671–680 (1983 May).

[4] G. B. Sorkin, "Efficient Simulated Annealing on Fractal Energy Landscapes," *Algorithmica*, vol. 6, pp. 367–418 (1991).

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, MI, 1975).

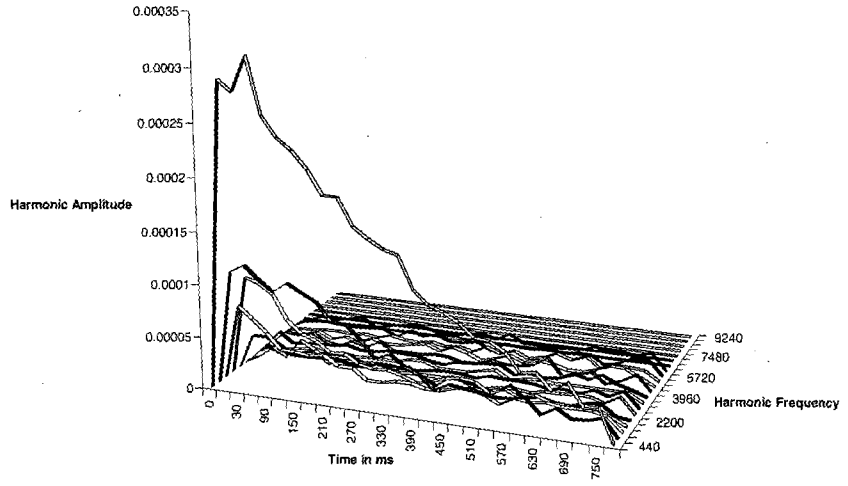


(a)

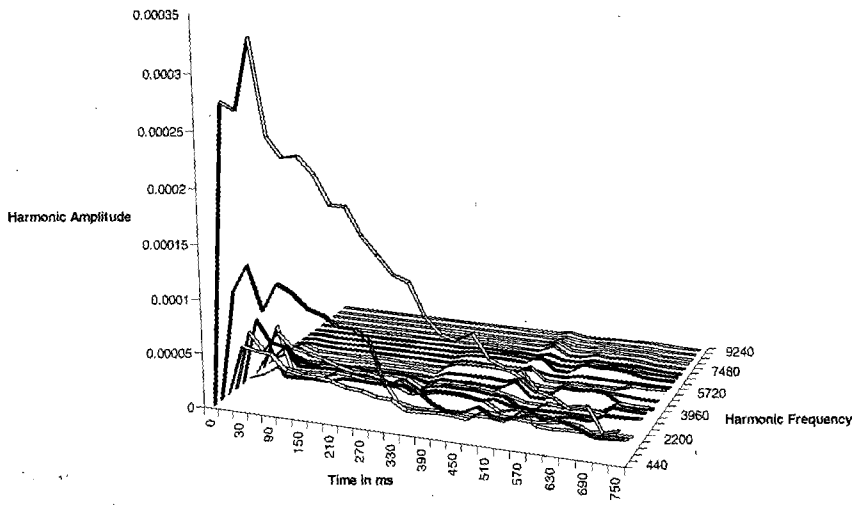


(b)

Fig. 17. Spectrum plots. (a) Sample saxophone at 440 Hz. (b) Synthesized saxophone at 440 Hz. Average fitness = 0.0149.

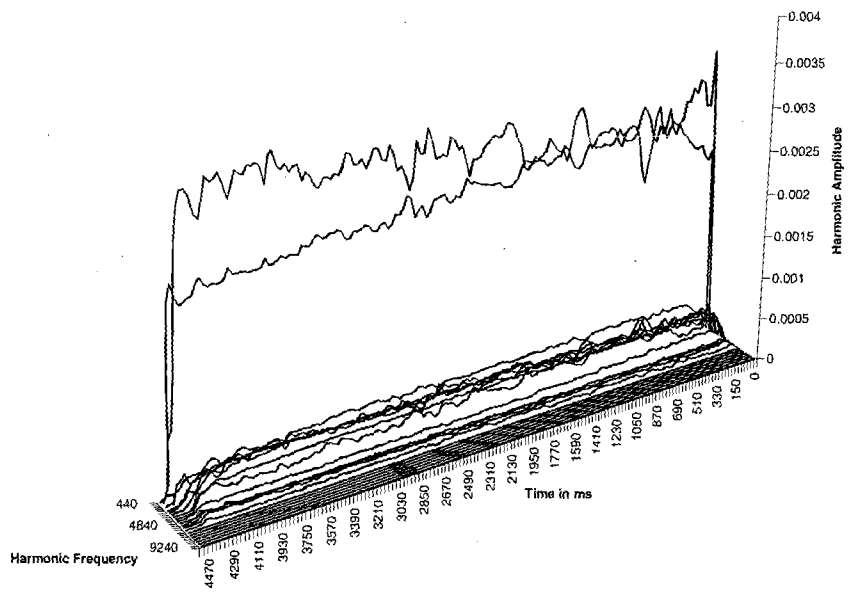


(a)



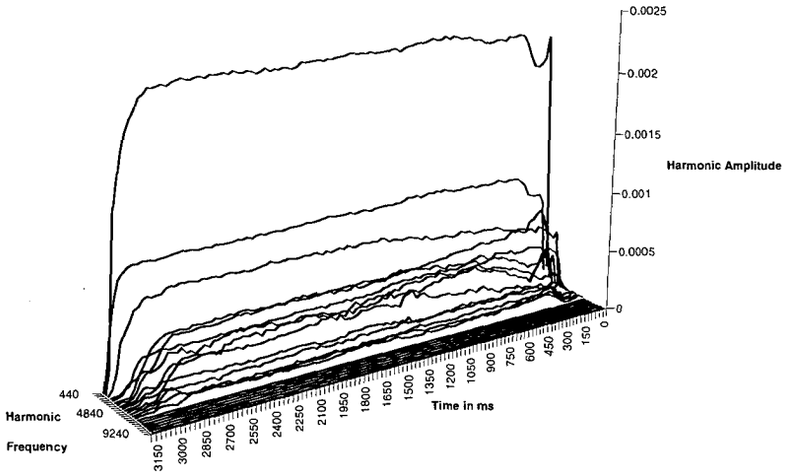
(b)

Fig. 18. Spectrum plots. (a) Sample piano at 440 Hz. (b) Synthesized piano at 440 Hz. Average fitness = 0.0234.

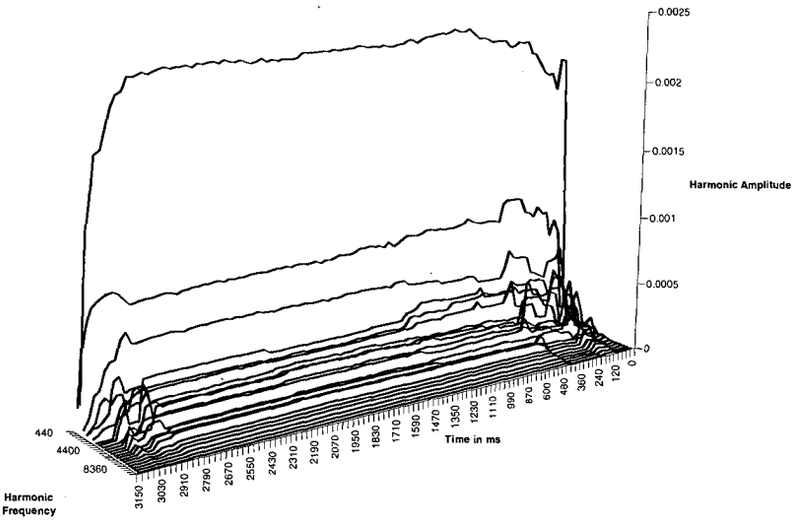


(a)

Fig. 19. Spectrum plots. (a) Sample oboe at 440 Hz. (b) Synthesized oboe at 440 Hz. Average fitness = 0.0283.

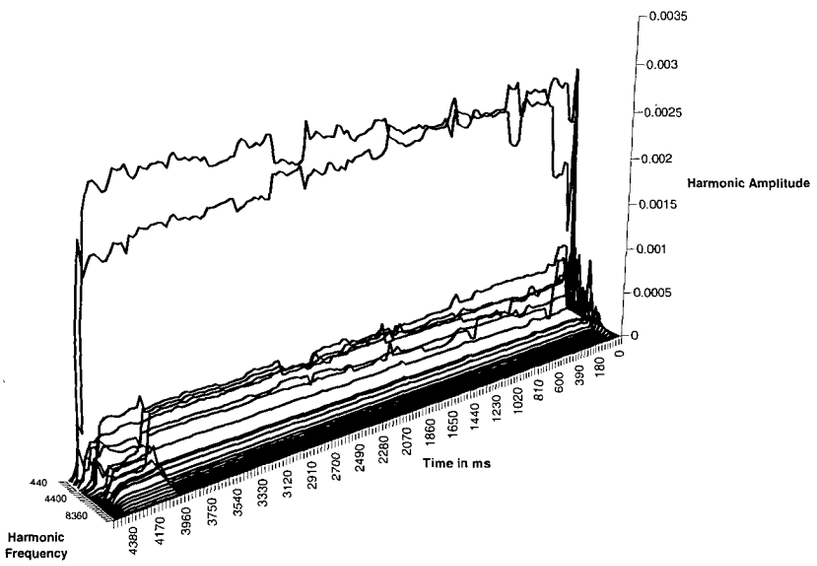


(a)



(b)

Fig. 20. Spectrum plots. (a) Sample trumpet at 440 Hz. (b) Synthesized trumpet at 440 Hz. Average fitness = 0.0139.



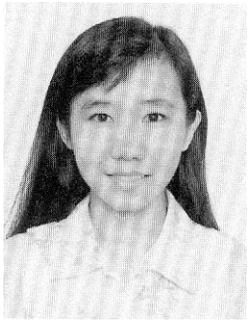
(b)

Fig. 19. Continued.

[6] L. Booker, "Improving Search in Genetic Algorithms," in *Genetic Algorithms and Simulated Annealing*, L. Davis, Ed. (Morgan Kaufmann, Los Altos, CA, 1987), pp. 61–73.

[7] A. Horner, J. Beauchamp, and L. Haken, "Machine Tongues XVI: Genetic Algorithms and Their Applications to FM Matching Synthesis," *Computer Music J.*, vol. 17, pp. 17–29 (1993 winter).

THE AUTHORS



S. M. Lim

S. M. Lim obtained a B.Sc. (Hons) in physics in 1994 from the National University of Singapore. She is pursuing a Ph.D. in the field of digital sound synthesis at the National University of Singapore. She is currently working at the Centre for Signal Processing on voice-based biometrics projects. Her research interests include digital sound synthesis, voice-based biometrics, genetic algorithms, and simulated annealing applications.

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