Genetic Algorithms Summary

Shi Yong
yong.shi2@mail.mcgill.ca

1. Introduction

Briefly to say, Genetic Algorithms (GA) is a searching technique inspired by Darwin's evolution theory. It was invented and developed by John Holland from the University of Michigan in the early seventies, and have been widely used in different areas, including computer music and audio technology.

GA can be used to efficiently solve complex optimization problems [1-3], especially those with huge search space and surprisingly complexity that are hard or even impossible to be solved by traditional minimum-seeking algorithms. It is especially useful when the analytical expression of the problem is unknown. Thanks to its self-evolving characteristic, GA can be used not only to solve problems in terms of traditional meaning, but also feasible to generate new schemes, such as evolving novel topologies of sound synthesis system[4], or automatic programming[1]. Another interesting feature is its interactivity, i.e. the fitness can be evaluated by human beings interactively and subjectively, rather than computed by a predefined fitness function running in computer[3]. This feature makes it possible and feasible to use GA creatively in music composing, visual art design, and new product design. Also, GA have been used in simulating the evolving of dynamic systems, such as ecosystems, immune systems, and social systems[1].

2. Description

GA borrows similar concepts from biological genetics and evolution theory, such as chromosome, gene, crossover, mutation, selection, fitness, etc. For a typical GA problem, it starts with a set of randomly selected solutions (called individuals), or population. An encoding scheme must be designed first such that all related parameters are mapped to the chromosome – which can be a string of binary bits or float point numbers, up to the nature of the problem. The individuals are distinguished from each other by the genetic code of their chromosome. The adaptability of an individual is evaluated by its fitness value. Generally, individuals with high fitness value intend to have higher survival probability and pairing opportunity. In the process from one generation to another, some individuals are selected as parents and paired (by means of predefined selecting and pairing methods), and then their genes are inherited by their offspring. New individuals are generated in principles of inheritance and variation, just like the case in the natural world, and new chromosome patterns (new solutions) appear in the new generation. To prevent premature convergences, the chromosomes of a few individuals are disturbed by mutation. After selecting and discarding, the survived offspring and parents make up the new generation while the population size keeps the same. The iteration continues until a convergence requirement is met, that means either an acceptable solution has been found or a predefined loop number is exceeded.

3. Customization Methods

The basic idea of GA is simple. Nonetheless, in order to tailor GA to a specific problem, a wide variety of techniques have been developed, that concerning different aspects such as chromosome encoding (mapping strategies), fitness function designing, crossover methods, mutation methods, selection and pairing strategies. Also, various GA parameters, such as population size, crossover probability, mutation probability, etc., must be taken into account in that the capacity of GA as solving a specific problem, such as accuracy and efficiency, is greatly affected by them. Some typical methods [2, 3]are listed below (not a complete list).

- Encoding:
  - Binary Encoding: Chromosome is encoded as a string of binary bits (1 or 0).
  - Permutation Encoding: Chromosome is encoded as a string of integers, which represents the order in a sequence. Useful for ordering problems, such as Traveling Salesman Problem.
  - Value Encoding: Chromosome is encoded as a string of real values, which can be anything connected to the problem, i.e. the parameters of an audio effect algorithm.
  - Tree Encoding: Chromosome is encoded as a tree of some objects, such as a function block in a topology structure. Mainly used in genetic programming.
- Crossover:
  - Single Point Crossover: Chromosome is divided to two parts, i.e. a1a2 + b1b2 => a1b2
  - Two Point Crossover: Chromosome is divided to three parts, i.e. a1a2a3 + b1b2b3 => a1b2a3
  - Uniform Crossover: Some bits are randomly copied from one parent.
  - Tree Crossover: In Tree Encoding, parents' trees are divided in one point.
- Mutation:
  - Bit Inversion: Some bits of the string are flipped
  - Adding: A small number is added or subtracted to the value, used for Value Encoding.
- Selection and Pairing:
  - Ranking and Pairing from top to bottom: This the easiest way
  - Random Pairing: Parents are selected randomly
  - Weighted Random Pairing: Probability of selection is weighted by fitness
  - Tournament Selection: A small subset of chromosomes are randomly picked, the chromosome with high fitness is selected, the process repeats for every parent needed
  - Blending Method: A single offspring variable value comes from the combination of two parents' variable values with a random interpolation factor, used for Value Encoding.

4. Applications in Music Technology

GA have been successfully used in music technology area, such as:

- Granular Synthesis Regulation [5]: Each grain's parameters (freq, amp, etc.) mapped to a chromosome's bit string. GA parameters (crossover rate, mutation rate, etc.) control the change in the grain population.
- Automatic Generation of Sound Synthesis [4]: Not only the parameters of synthesis, but the synthesis building blocks are regarded as genes. Generates new topologies and synthesize the target sound.
- Parameters Optimization for Natural Sounding Synthesis [6]: GA with real value encoding is used to implement an automatic parameter extraction method for natural string synthesis. Fitness is computed from a comparison of the perceptual transformed spectra between synthesized sound and target sound by making use of the nature of human aural perception.
- FM Synthesis Parameters Optimization [7]: FM parameters (carrier freq, mod freq, amp) mapped to a chromosome’s bit string. Spectrum similarity of target sound and synthesized sound is evaluated as fitness.
- Synthesis Control to Mimic Target Sound [8]: GA was used to drive a software synthesizer to solve the target sound matching problem. Fitness function was defined in sound attributes matching. A layered learning method was used to improve the performance.
- Synthesis of Low Peak Amplitude Signals [9]: GA is used in designing low peak amplitude signals by exploring many local optima of the phase space in parallel.

References

1. Forrest, S., *Genetic algorithms: principles of natural selection applied to computation.* Science, 1993. **261**(5123): p. 872-878.
2. Obitko, M. *GENETIC ALGORITHMS*. [cited 2007 Jan 30]; Available from: http://cs.felk.cvut.cz/~xobitko/ga.
3. Haupt, R.L. and S.E. Haupt, *Practical Genetic Algorithms*. 2nd Edition ed. 2004, Hoboken, New Jersey: Wiley-Interscience. 272.
4. Garcia, R.A., *Towards the Automatic Generation of Sound Synthesis Techniques: Preparatory Steps*, in *Audio Engineering Society 109th Convention*. 2000: Los Angeles.
5. Fujinaga, I. and J. Vantomme. *Genetic Algorithms as a Method for Granular Synthesis Regulation*. in *International Computer Music Conference*. 1994.
6. Riionheimo, J. and V. Valimaki, *Parameter Estimation of a Plucked String Synthesis Model Using a Genetic Algorithm with Perceptual Fitness Calculation.* EURASIP Journal on Applied Signal Processing, 2003(8): p. 791-805.
7. Lai, Y., et al., *Automated Optimization of Parameters for FM Sound Synthesis with Genetic Algorithms*, in *International Workshop on Computer Music and Audio Technology*. 2006.
8. McDermott, J., N.J.L. Griffith, and M. O'Neill. *Target-Driven Genetic Algorithms for Synthesizer Control*. in *9th Int. Conference on Digital Audio Effects (DAFx' 06)*. 2006. Montreal, Canada.
9. Horner, A. and J. Beauchamp, *A genetic algorithm-based method for synthesis of low peak amplitude signals.* Journal of the Acoustic Society of America, 1996. **99**(1): p. 433-443.