Designing Like Mother Nature

An Introduction to Genetic Algorithms

Dr. Derek S. Linden
(703) 561-3400 dlinden@ieee.org

About the Speaker, Derek Linden

- M.S., MIT, 1993, EE (Solid State Devices/Superconductivity)
- Rome Lab, 1993 - 1996, basic research on superconductors at microwave frequencies, antennas, GAs
- Current research: Increasing GA efficiency, applying GAs to new problems
- Linden Innovation Research LLC
  - Automated Design and Optimization Consulting, Training and Software
Introduction to GAs: Overview

• Goal: to introduce the fundamental concepts of a GA
  – What is a GA?
  – GA basics
  – Examples

What is a GA?

• A probabilistic, iterative search and optimization strategy
• Mimics biological intra-species adaptation and evolution through mating and survival-of-the-fittest
• Finds optima for many types of numerical problems
• Requires:
  – A coding strategy
  – An objective function
  – A mating and mutation scheme
The GA Iterative Process

- Initialize new population
- Simulate and evaluate new members
- Mutate children
- Choose mates and create children
- Rank-order all members
- Is convergence criteria met?
  - YES: Output results
  - NO: Repeat

GA Terms

- Genes
- Alleles
- Population
- Chromosomes

- Alleles:
  - 0 0 1 1 0 1 0: 0.546 0.010 0.530 0.223 0.750 0.456
  - 0 0 1 1 0 0 1 1 0 0: 0.754 0.122 0.822 0.564 0.438 0.990
  - 0 1 1 0 1 1 0 0 1: 0.945 0.678 0.800 0.442 0.901 0.198
  - 1 0 1 0 1 1 0 0 1 0: 0.248 0.548 0.401 0.881 0.058 0.451
  - 1 1 0 0 1 1 0 0: 0.700 0.890 0.540 0.111 0.878 0.002
An Example Design Problem

4 Variables, with Constraints:
- Material: ceramic, glass, plastic
- Diameter: 2”-5”
- Height: 3”-6”
- Thickness: 0.1”-0.5”

Dependent constraint:
- Weight < 1.5 lbs.

Optimize for:
- Heat Retention = f(M,d,h,t)
- Cost = f(M,d,h,t)
- Volume = f(d,h)

Setup for GA Optimization

Chromosome:

<table>
<thead>
<tr>
<th>Material</th>
<th>Diameter</th>
<th>Height</th>
<th>Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>4.0”</td>
<td>2.8”</td>
<td>0.447”</td>
</tr>
</tbody>
</table>

Objective Function =

Heat Retention + Volume - Cost - Penalty - Weight
(Penalty is non-zero only if Weight above 1.5 lbs.)
Snapshot During the GA Process

Brief History

• Before the GA, algorithms based on mutation were tried

• John Holland (University of Michigan)
  – Holland had the basic GA by the mid-1960s
  – Monograph in 1975—“Adaptation in Natural and Artificial Systems”
  – Purpose: to understand adaptive processes in natural systems and design artificial systems that mimic natural system behavior

• David Goldberg—textbook in 1989
Current Areas of Application

- Mechanical Engineering
- Software Design
- Electromagnetics
- Electrostatics
- Artificial Intelligence/Artificial Life
- Robotics
- Aeronautical Engineering
- Financial

The GA Process

- Set up simulator/equations to evaluate members of population
- Define problem—constraints, unknowns, variables
- Determine objective function
- Determine chromosome mapping
- Determine genetic algorithm characteristics
  - mating selection, crossover, mutation, population size, etc.
- Run the GA optimization process
- Output the optimal design characteristics
Objective Function

- Gives a single score based on simulation results
- Used to rank-order the members of the population
- Single criteria or multi-criteria
- Include any penalty terms for violating constraints

\[
\text{Fitness} = -c_1 \cdot \text{gain} + c_2 \cdot \text{mismatch} + c_3 \cdot \text{distortion} \\
+ c_4 \cdot (\text{amount of power violation})
\]

1-D Binary and Real Chromosomes

- Binary:
  0 0 1 1 1 0 1 0 1 0
  - Usually each variable consists of several bits
  - Most commonly used by far, good for most problems

- Real:
  0.546 0.010 0.530 0.223 0.750 0.456 0.555
  - Usually each variable consists of only one number
  - Use for problems involving mostly real, continuous variables
Chromosome Mapping: Example

Mating Process

- The basic mating process:
  - Eliminate poor performers (total population remains constant)
  - Choose chromosomes to mate
  - Create offspring
- Simple GA: replaces whole population with new children, though some are copies of parents
- Steady-State GA: saves a portion of the population each generation
- Elitist: saves top chromosome
In biology, mates are chosen through natural selection
- Brightest flower, strongest male, most attractive call
- Most common GA method: weighted roulette wheel

Usually weighted by fitness, or qualities like similarity
1-D Binary Mating—Single-Point Crossover

- Parent chromosomes:
  - [00011110]
  - {11001100}
- Let the crossover point be between the 5th and 6th bit (but could be between any two bits)
- Children: [00011]{100}
  - {11001}{110}
- Works the same way for real chromosomes, except no functional genes are able to be split

1-D Real Chromosome Mating

- Heuristic crossover

<table>
<thead>
<tr>
<th>Fitness</th>
<th>Gene Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Child</td>
<td>X</td>
</tr>
</tbody>
</table>

- Quadratic crossover

<table>
<thead>
<tr>
<th>Fitness</th>
<th>Gene Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Child</td>
<td>X</td>
</tr>
</tbody>
</table>

- Many other methods exist

Adewuya, 1996
1-D Binary and Real Chromosome Mutation

• Binary: Bit flip
  – Flip a randomly selected 1 → 0 or 0 → 1

• Real: Uniform mutation

• Real: Gaussian mutation

The GA Parameters

• Population Size: 30 - 10,000
  (most I’ve heard of: 1,000,000)

• Parent pool size (overlap): 10%-50%

• Probability of mutation: < 2%

• Convergence criteria:
  – # generations
  – # simulations
  – non-improvement
  – loss of diversity
  – when I choose to stop it
Putting the Process Together

- Initialize new population
- Simulate and evaluate new members
- Mutate children
- Choose mates and create children
- Rank-order all members
- Is convergence criteria met?
  - YES: Output results
  - NO: Repeat

Typical GA Behavior: Fitness

- Best fitness, Average fitness vs. Generation

![Fitness vs. Generation Graph]
GA Advantages

- Properly implemented, it can lead to optimal solutions relatively rapidly and efficiently
- Prevents the solution from getting trapped in local minima through parallelism
- Is zeroth-order/blind—requires no information other than the objective function value for each chromosome
- Can optimize very complicated systems with no human intervention (not even an initial guess!)
- Very robust to parameters, coding, etc.
- Able to be implemented in a parallel manner

GA Examples

- Discrete problems
  - Truss topology design
  - VLSI connection design
  - Job Shop Process Planning
- Continuous problems
  - Turbine engine design
  - Pattern nesting (Parts layout)
  - Simple wire antenna
  - Folded monopole & Crooked-wire antennas
Truss Topology Design

- Use a GA to determine an optimal truss structure with the least amount of material given a load

Chapman et al., 1994

Truss Topology Design

- Example optimized designs at differing resolutions

(a)  
(b)  
(c)
Job Shop Process Planning

- Minimize the cost and hassle in machining custom parts
- Many different combinations of machine, tool, and setup are possible to create the same part

Zhang, et al., 1997

<table>
<thead>
<tr>
<th>Minimize</th>
<th>Machine</th>
<th>Setup</th>
<th>Tool</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
<td>0</td>
<td>11</td>
<td>13</td>
<td>2664</td>
</tr>
<tr>
<td>Setup</td>
<td>8</td>
<td>0</td>
<td>10</td>
<td>3799</td>
</tr>
<tr>
<td>Tool</td>
<td>12</td>
<td>5</td>
<td>0</td>
<td>5014</td>
</tr>
<tr>
<td>Cost</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>1739</td>
</tr>
</tbody>
</table>
VLSI Connection Design

- Rather complex GA technique compared with 2-4 other standard VLSI techniques in each of 10 classic benchmarks
- GA was best method for each benchmark in numbers of vias
- Best for overall wire length in 7 of 10 benchmarks
- 2nd best in the other 3 benchmarks, and usually a close second
- Crosstalk requirements can be added, which none of the other techniques can handle

Leinig, 1997

Turbine Engine Design

At least 100 variables, each with a continuous range

Search space of $10^{387}$ points

Fitness: compliance with about 50 constraints + performance measures

Engineer: 8 weeks for a satisfactory design
Engineer + Expert system: less than 1 day w/ 2x improvement
GA + Expert system: 2 days w/ 3x improvement over engineer alone

Holland, 1992
Pattern Nesting

• Applications in many industries
  – Clothing
  – Shipbuilding
  – Automobile part manufacturing

Dighe & Jakiela, 1996

Pattern Nesting

• Minimizing rectangular enclosure
  68.4%
  69.0%

(c) Linden Innovation Research LLC
Pattern Nesting

- Minimizing height

70.4% 72.4%

Simple Wire Antenna

- The design

Driven element 0.5 \( \lambda \)

Drive point (in center of element)

Separation distance 0.04 - 2 \( \lambda \)

Reflector element 0 - 4 \( \lambda \)

Linden, 1997
Simple Wire Antenna

- The search space

![3D graph showing the relationship between length, separation, and gain]

Simple Wire Antenna

- The objective function
  - Maximize gain in forward direction (already a single number)

- The chromosome
  - Two real values for length and separation

- GA parameters
  - 20 chromosomes, 50% overlap, 0.6% mutation
Simple Wire Antenna

Folded Monopole, Crooked-Wire Antennas

- The problem: Our goal in each case was to achieve a single objective: the broadest beam possible over the upper hemisphere

\[
\text{Score} = \sum_{\theta, \phi} \left( \text{Gain}(\theta, \phi) - \text{Avg. Gain} \right)^2
\]

- Folded monopole — power gain only
- Crooked wire antennas — RH circular polarization gain
The Folded Monopole Chromosome

Goal: Hemispherical coverage, regardless of polarization

Folded Monopole Results

Altshuler & Linden, 1997
Crooked-Wire Genetic Antenna

Space

Goal: Coverage over hemisphere 10 above the horizon with right-hand circular polarization

(X1,Y1,Z1)(X2,Y2,Z2)(X3,Y3,Z3)...(X7,Y7,Z7) 105 bits

Crooked-Wire Genetic Antenna

Results

Gain (dB)

θ(deg.)

f = 1600MHz

φ=0° 135°

45° 90°

Linden & Altshuler, 1996
# References