Genetic Algorithms

Optimization problems and genetic programming
Optimisation 101

- Calculus-based
  - Direct (hill-climbing)
  - Indirect
- Enumerative
- Stochastic
  - Stochastic hill-climbing
  - Simulated annealing
  - Genetic algorithms

D.E. Goldberg,
GA Kangaroos

- Highest peak in Himalaya?
- Parachute a lot of kangaroos at random places
- Kangaroos wander and create more kangaroos
- Each generation, remove low-altitude kangaroos
- At the very end, there are only kangaroos on Mt. Everest and (perhaps) on Mt. K2
Theory of Evolution 101

- Britannica Enciclopaedya says...
  “theory in biology postulating that the various types of plants, animals, and other living things on Earth have their origin in other preexisting types and that the distinguishable differences are due to modifications in successive generations...”

... and continues with a long article by Prof. Francisco J. Ayala (U. California, Irvine)
Theory of Evolution 101

- After the discovery of DNA the mechanism of species evolution can be explained in terms of species and genes.
- Species are encoded by their genes and genes are the ones who fight for surviving in the gene pool by means of associating to other genes to develop successful survival machines.

Evolution & Computers

- It is straightforward to simulate (toy) evolution in a computer
- We define individuals with a set of characteristics (parameters, genes)
- We evolve those characteristics, generation after generation against an environment (some genes survive, some genes do not)
- But then... we are optimising the population to the environment...
Mathematical POV of Evolution

- Evolution IS an optimisation scheme
- The different kinds of species evolve to the optimal adaptation to the surrounding environment. Thus, evolution is an “algorithm” that searches for the best solution creating a set of individuals (a generation), it decides which individuals are the best ones, and, by means of crossover, keeps the good genetic characteristics for the next generation -that will be closer to the optimal solution- and removes the individuals with worst genetic content
Biology vs Optimisation

- Environment ↔ Objective function
- Individual ↔ Set of parameters
- Generation ↔ Set of solutions
Introduction (I)

- Inspired by natural evolution
- Population of individuals
  - Individual is feasible solution to problem
- Each individual is characterized by a fitness function
- Based on their fitness, parents are selected to reproduce offspring for a new generation
  - Fitter individuals have more chance to reproduce
  - New generation has same size as old generation; old generation dies
Introduction (II)

- Offspring has combination of properties of two parents
- If well designed, population will converge to optimal solution
Algorithm

BEGIN

Generate initial population
Compute fitness of each individual
REPEAT /*new generation/*
Generate new offspring
Compute fitness
UNTIL population has converged
END
Introduction (III)

- Reproduction mechanism has no knowledge of the problem to be solved
- Link between genetic algorithm and problem:
  - Coding
  - Fitness function
Basic principles (I)

- Coding or representation
  - String with all parameters
- Fitness function
  - Parent selection
  - Scaling
- Reproduction
  - Crossover
  - Mutation
- Convergence
  - When to stop
Basic principles (II)

- An individual is characterized by a set of parameters: Genes
- Genes joined into a string: Chromosome
- Reproduction is a “dumb” process
- Fitness is measured in the real world “struggle for life”
Particularities of GAs

- Whereas most methods employ a single solution which evolves to reach the local optimum, GAs work on a population of many possible solutions simultaneously.
- GAs only need the objective function to determine how fit an individual is. Neither derivatives nor other auxiliary knowledge are required.
- GAs use probabilistic rules to evolve (randomness does not mean directionless!)
Coding

- Parameters of the solution (genes) are concatenated to form a string (chromosome)
- All kind of alphabets can be used for a chromosome (numbers, characters), but generally a binary alphabet is used
- Order of genes on chromosomes can be important
- Good coding is probably the most important factor for the performance of a GA
- In many cases, many possible chromosomes do not code for feasible solutions
TSP coding

- **Binary**
  - Cities are binary coded; chromosome is a string of bits
    - Most chromosome code for illegal tour
    - Several chromosomes code for the same tour

- **Path**
  - Cities are numbered; chromosome is a string of integers
    - Same problems

- **Others**
Next generation

- **Fight**
  - There is a chance that the chromosomes of the two parents are copied unmodified as offspring

- **Crossover**
  - Two parents produce two offspring
  - There is a chance that the chromosomes of the two parents are randomly recombined (crossover) to form offspring
  - Generally the chance of crossover/win is between 0.6 and 1.0

- **Mutation**
  - There is a chance that a gene of a child is changed randomly
  - Generally the chance of mutation is low
Crossover

- **One-point crossover**
  - Select one random point

- **Two-point crossover**
  - Select two random points

- **Uniform crossover**
  - Generate mask

- **Arithmetic crossover**
Problems with crossover

- Depending on coding, simple crossovers can have high chance to produce illegal offspring
  - TSP
- Uniform crossover can often be modified to avoid this problem
  - Where mask is 1, copy cities from one parent
  - Where mask is 0, choose the remaining cities in the order of the other parent
Fitness function

- Purpose
  - Parent selection
  - Measure for convergence
  - Selection of individuals to die
- Should reflect the value of the chromosome in some “real” way
- Next to coding the most critical part of a GA
Parent selection

- Chance to be selected as parent proportional to fitness
  - Roulette wheel
- To avoid problems with fitness function
  - Tournament
GA Flow Diagram

Initial generation (random)

Scale population and set survival and mating probabilities

Fight  Crossover

New population  Mutate

Convergence?

No  Yes

Solution!
Problems with fitness range

- Premature convergence
  - $\Delta$ Fitness too large
  - Relatively superfit individuals dominate population
  - Population converges to a local maximum
  - Too much exploitation; too few exploration

- Slow finishing
  - $\Delta$ Fitness too small
  - No selection pressure
  - After many generations, average fitness has converged, but no global maximum is found; not sufficient difference between best and average fitness
  - Too few exploitation; too much exploration
Solutions

- Tournament selection
  - Implicit fitness remapping
- Adjust fitness
  - Fitness scaling
  - Fitness ranking
Fitness scaling

- Fitness values are scaled by subtraction and division so that worst value is close to 0 and best value is close to a certain value typically 2
  - Chance for the most fit individual is twice the average
  - Chance for the least fit individual is close to 0
- Problems when the original maximum is very extreme (super-fit) or when the original minimum is very extreme (super-unfit)
  - Can be solved defining a minimum and/or a maximum value for the fitness
Fitness ranking

- Individuals are numbered in order of increasing fitness
- The rank in this order is the adjusted fitness
- Starting number and increment can be chosen in several ways and influence the results

- No problems with super-fit or super-unfit
- Often superior to scaling
Mutation

- Mutation rate
- Allows to explore areas not explored by crossover
Other parameters (I)

- Initialization
  - Population size
  - Grain

- Reproduction
  - Generational
  - Generational with elitism
  - Steady state: two parents reproduce and two parents die
Other parameters (II)

- **Stop criterion**
  - Number of new chromosomes
  - Number of new and unique chromosomes
  - Number of generations

- **Measure**
  - Best individual
  - Average

- **Duplicates**
  - Accept duplicates
  - Avoid many duplicates
  - No duplicates at all
Typical best individual evolution

\[ \frac{\chi^2}{\chi^2_{\text{min}}} \] (logarithmic scale)

Generation
Dumb example: $f=\text{asin}(x)$

**Fight:** $X_{32}$ vs $X_{55}$

Flip a coin, the winner goes to next generation

**Crossover**

$X_{70} = [34567890]$
$X_{24} = [10950322]$

**Offspring**

$X_{\text{new}} = [34560322]$

**Mutation**

$X_{\text{new}} = [34560322]$

$X_{\text{mutated}} = [34560122]$
Mixed strategy

- Genetic + Hillclimbing
- Faster
- Hillclimbing gains accuracy
Multi-objective: Pareto fronts

- Two (or more) objective functions
Author: P.M.G. Corzo
Artificial Intelligence

- Video games
  - Shoot’em up
  - Bots in Quake III Arena
Change of Paradigm: Symbolic Regression

- We expand the concept of optimisation
- It is not just to fit parameters to a given function
- The function itself is part of the problem

- [One step further: the optimisation technique can be part of the problem too]
Biology vs Optimisation

- Environment ↔ Objective function
- Individual ↔ Set of parameters & function
- Generation ↔ Set of solutions
Distilling Free-Form Natural Laws from Experimental Data

Michael Schmidt
Hod Lipson

Cornell University
Cornell Computational Synthesis Lab