Slide 1

Part I — The Essentials of Genetic Algorithms

1. What are genetic algorithms and why are genetic algorithms used in A.I.?
2. A simple genetic algorithm
3. Extentions to the simple genetic algorithm
4. How genetic algorithms work — an introduction to theory

Slide 2

Genetic Algorithms

References:

- “Adaptation in Natural and Artificial Systems”, J. H. Holland, 1973 (2nd edition 1992).
- “Genetic Algorithms in Search, Optimization, and Machine Learning”, D. E. Goldberg, 1989.
- “An Introduction to Genetic Algorithms”, M. Mitchell, 1996.

Slide 3

What are genetic algorithms?

They are *search* algorithms.

- They act on a *population* of candidate solutions.
- Changes to the constituents of the population occur through random changes.
- The better candidates remain in the population; the worse ones are discarded.
- Modelled on directed changes in a biological population — breeding, forced molecular evolution, natural selection, etc.

An example of an *evolutionary computation* method.

Slide 4

Why are genetic algorithms used in artificial intelligence?

Because *search* is important operation in artificial intelligence.

Reasoning tasks: Search for an answer to a problem made difficult by hard and soft constraints. E.g.

- Find a schedule which satisfies a large number of people.
- Find a set of moves which results in a given final state.
- etc.

Genetic algorithms are used for soft constraint satisfaction, scheduling problems, finding game strategies, and so forth.

Slide 5

Machine learning: Choose a model from a class of models which solves a problem by interacting with the problem.

One approach is to *search* over the class of models to find the one which performs well on the task. Genetic algorithms are a useful when:

1. The search space is discrete.
2. The learning task is a reinforcement learning task.
3. There is a natural use for a population of solutions.
4. There is a desire to parallelise the search.

Similar to *generate-and-test* algorithms: simulated annealing and stochastic hill-climbing.

Slide 6

Historical Aside

Reference: Evolutionary Computation: The Fossil Record, David B. Fogel, editor, 1998.

Genetic Algorithms: (Holland, 1973; Goldberg, 1989)

Representation: Strings of characters

Operators: Crossover taken as very important, mutation.

Evolution Strategies: (Rechenberg, 1965; Schwefel, Back, Hoffmeister)

Representation: real numbers (parameters)

Operators: mutation = Gaussian noise, no crossover.

Evolutionary Programming: (L. Fogel, et. al. "Artificial Intelligence through Simulated Evolution", 1966).

Representation: Finite-state automata.

Operators: Various forms of mutation

Genetic Programming: (Koza, 1990)

Slide 7

Representation: Parse trees of LISP expressions

Operators: Like genetic algorithms.

Slide 8

Collectively, these are called Evolutionary Computation.

What they share:

- Population-based dynamics.

How they differ:

- How the problem is represented.
- Relative importance of the various dynamical operators.

Slide 9

Elements of Genetic Algorithms

Representation: Potential solutions to the problem represented as fixed length strings of characters drawn from some alphabet. String length L.

a b b a b c a b,

1 0 0 1 1 1 0,

etc.

(Sometimes strings of real values are used).

Population: fixed size, N, collection of strings.

Objective function: A function of the strings, $E(x)$, a quantity to be optimised.

Slide 10

Dynamics: The population is changed by the action of three operators:

Selection: This operator selects strings from the current population such that better strings are more likely to be chosen.

Recombination/Crossover: This operation produces offspring from a pair of parents by combining parts of each parent. Each offspring contains part of the strings of each parent.

Mutation: This operation simply changes characters in the offspring string at random. Each character is changed with a small probability.

Slide 11

A Simple Genetic Algorithm

There are a number of different variations. Here I show the simple form.

Dynamics:

```
current_population=N random strings.
repeat
   repeat
      Select 2 strings from current population using selection operator.
      Produce 2 offspring using recombination.
      Mutate those 2 offspring.
      Add the 2 offspring to new_population.
   until size_of(new_population) = N.
   current_population = new_population.
until (stopping criterion met)
```

Slide 12

Fitness Proportional Selection — Introduce a *Fitness Function* $F(x)$, positive and increasing with better solutions.

Select string α with probability (with replacement)

$$p^\alpha = \frac{F^\alpha}{\sum_\gamma F^\gamma}$$

where F^α is the fitness of the string α.

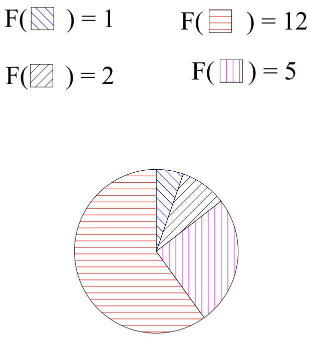
Examples of fitness functions:

$$F^\alpha = E^\alpha$$

$$F(E^\alpha) = \exp(sE^\alpha).$$

Slide 13

Roulette Wheel Metaphor:



Slide 14

Mutation Changes the value of each gene with independent probability p_m; leave it unchanged with probability $1 - p_m$. Typically $p_m \sim 1/L$ or some small value.

Crossover Crossover works on pairs of strings. For each pair crossover occurs with probability p_c and with probability $1 - p_c$ the offspring are identical to the parents. Typical value is $p_c \approx 0.6$.

Slide 15

Single-point crossover: Choose a crossing site at random.

parent 1:	$1111111 \vert 1111111$	offspring 1:	$0000000 \vert 1111111$
parent 2:	$0000000 \vert 0000000$	offspring 2:	$1111111 \vert 0000000$

Multi-point crossover:

parent 1:	$111 \vert 111111 \vert 11111$	offspring 1:	$000 \vert 111111 \vert 00000$
parent 2:	$000 \vert 000000 \vert 00000$	offspring 2:	$111 \vert 000000 \vert 11111.$

Uniform crossover: Each site is independently chosen from parent 1 or parent 2.

Slide 16

Why this might work

1. Parallel hill-climbers — can explore multiple peaks.
2. Crossover makes big (non-local) changes.
3. Combining partial solutions can produce better solutions (the building block hypothesis).

Slide 17

Why this might fail to work

1. "Premature convergence" — the population can become very similar. Then
   - crossover does nothing
   - becomes a very inefficient hill-climber, as it must drag the population around with it.
2. Crossover can be very disruptive if the characters in the string interact strongly in producing the performance.

Slide 18

Example — Colouring a Map

Task: Colour a map so neighbouring countries are coloured with different colours.

Representation: A string of colours; each character representing a different country.

Objective function: The number of borders between countries which are differently coloured. Maximise this.

Fitness function: Boltzmann selection,

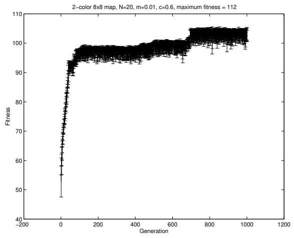
$$F = \exp sE.$$

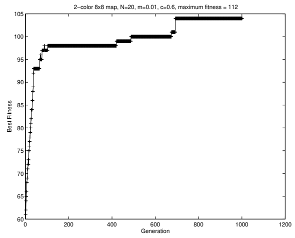
Crossover: Single-point.

Mutation: Change the colour of a country to a randomly chosen different colour.

Slide 19

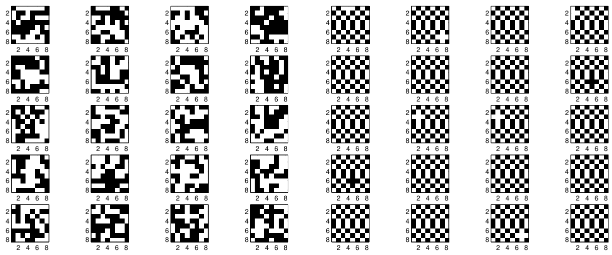
Two-colouring a square map





Evolution of the mean and variance　　evolution of the best

Slide 20



Initial population　　Final population

Population shows premature convergence.

Slide 21

Extensions and variations of the simple genetic algorithm

Slide 22

Alternative Selection Models

To avoid premature convergence, other forms of selection are often used.

Island models and niching: Population is divided into subgroups. Competition is within subgroup only. Some migration occurs.

Fitness sharing: Fitness is shared between strings to a degree determined by their similarity.

Steady-state selection: Only a small part of the population changed each time step (as opposed to "generational selection").

Elitism: The best members of the population are selected deterministically.

Slide 23

Alternative Crossover Operators

Many problems have constraints which standard crossover would break.

Example: Travelling Salesman Problem. A tour is a permutation.

Tour 1: $325|41$

Tour 2: $123|54$

$$\text{Crossover yields} \begin{cases} 32554 \\ 12341 \end{cases}$$

Not valid tours!

Operators should reflect the structure of the problem.

Slide 24

Dealing with constraints

Often applied to a task with constraints, e.g. optimize

$$f : S \to R$$

such that a solution $x \in S$ satisfies certain inequalities or equalities.

Three approaches

1. Devise genetic operators which never produce infeasible solutions from feasible ones. (Generally felt to be the best approach)
2. Use a repair mechanism to fix infeasible solutions (Larmarkian or Darwinian?)
3. Introduce penalty terms to the objective function which penalize solutions which do not satisfy the constraint. (Widely used, because it is often difficult to figure out how to do the above).

Parallel Evolutionary Algorithms

Slide 25

Reference: Stender, *Parallel Genetic Algorithms* 1990

Genetic Algorithms parallelize well, since only selection requires communication accross the population, and some selection models require little communication.

Island models: can be implemented on coarse-grained parallel architectures. Each processor runs an independent GA, with occasional migration between generations (e.g. processors swap their best members).

Tournament selection: Each processor handles one tournament and the mutation and crossover of the participants.

Multi-objective function optimization

Slide 26

Reference: Fonseca and Fleming, ICGA-5, 1993

Often there are several objective functions, E_i, to optimise.

Examples:

- maximize profit but minimize ecological damage;
- maintain quality but reduce costs; etc.

Could optimize

$$E = \sum_i \alpha_i E_i$$

where α_i expresses the importance of each objective function. But the α's may be difficult to determine.

Use the population to explore the "surface of trade-offs".

Slide 27

Define: x dominates y if

$$\begin{aligned} f_i(x) &> f_i(y) \text{ for some } i, & (1)\\ &\text{and} & (2)\\ f_j(x) &\geq f_j(y) \text{for all } j. & (3) \end{aligned}$$

Let the fitness be the number dominated by a particular individual (so it depends upon the population). Useful to combine this with some form of niching to encourage diversity in the population. The result could be a population which is distributed over the non-dominated surface.

Slide 28

Example: Constrained Optimization of Gas Supply Network (Surrey, Radcliffe, Boyd. Control and Cybernetics, 1997.)

Task: minimize the cost of the pipes in a British Gas pipe network

constraints:

- pressure must not fall below a certain amount
- each pipe must have a pipe upstream of equal or greater diameter.

Treat the constraints as a set of objective functions; select a fraction on those objective fuction, and the remaining on cost.

Results: GA with penalty function and GA with multioptimization for constraints both reduced cost by 4% over existing heuristic method. Multi-objective function method reduced computational costs by removing need to search for parameters (α's from above).

Slide 29

Introduction to Genetic Algorithm Theory

Reference: Mitchell 1996, chapter 4.

Slide 30

How Genetic Algorithms Work — Some Ideas about Genetic Algorithm Theory

Questions theory might address:

1. On what types of problems will GAs work well?
2. How and why do they fail?
3. What causes punctuated dynamics and what determines waiting time between epochs?
4. Can their performance be improved or optimised?

Slide 31

Essential Features of a Useful Theory

Large Problem sizes — Number of possible strings $\gg$ run time.

Finite population sizes — Number of possible strings $\gg$ population size.

"Strong" selection — Analogous to artificial selection.

Realistic Recombination operators

Fitness landscape containing some aspects of realistic problems

Many theories use infinite population approximations (sometimes implicitly) and other unsatisfactory simplifications.

Slide 32

Limiting Cases

Infinite Population: (many authors)

- no fluctuations; everything takes its expected value.
- no search; every possible string is present in population.

Weak Selection: (Wright-Kimura, Derrida-Higgs) "genetic drift" (sampling fluctuations) are important sources of loss of diversity.

Asexual population (no crossover)

Very Sexual population: (Rattray, 1996) - no "linkage", everything described in terms of allele frequency.

$L = 1$**:** (Wright-Fisher Model)

$L = \infty$**:** (Woodcock and Higgs, 96; Prügel-Bennett, 96, Müller's Ratchet) No back mutations.

Slide 33

Holland - Goldberg Theory

Holland, 1975; Goldberg, 1989

Holland believed,

1. In natural adaptive systems, "epistasis" is important; some features must be present for the system to have any utility at all.
2. GAs are ideal for solving problems whose nature is, find the components that have to be present. "Building Block Hypothesis".

Schema — a template $\boxed{1\ 0 * 1\ 1\ 0 * * 0\ 1}$ where * could be anything, but the 1's and 0's must be present.

Slide 34

Schema Theorem

Let

$$\begin{aligned} m(H,t) &\equiv \text{number of instances of schema } H \text{ at time } t. && (4)\\ u(H,t) &\equiv \text{average (observed) fitness of schema } H \text{ at time } t. && (5)\\ d(H) &\equiv \text{length of schema.} && (6)\\ o(H) &\equiv \text{number of defined bits .} && (7)\\ \overline{f(t)} &\equiv \text{average population fitness at time } t. && (8) \end{aligned}$$

Then

$$E\left(m(H,t)\right) \geq \frac{u(H,t)}{\overline{f(t)}} m(H,t) \left[1 - p_c \frac{d(H)}{L-1}\right] \left[1-p_m\right]^{o(H)}. \quad (9)$$

Point: fitter than average, short, low-order schemata increase exponentially in time.

Slide 35

Allocation of Resources by the GA

Holland believed that the GA allocates resources to schemata optimally.

Viewed as a K-armed bandit.

- Each arm gives unknown payoff $p_i; i = 1 \dots k$.
- How should trials be allocated to arms such that
  1. it is learned which arm is best
  2. pay-out is maximised

  (exploration-exploitation dilemma).
- Answer: (for $k = 2, N$ trials) allocate $o(\log N)$ to apparently 2nd best arm.

Slide 36

Conclusion: for the 2-armed bandit, one must allocated exponentially more trials to the better arm; the GA allocates exponentially more population to fitter schema. Thus, it allocates resources optimally.

Implicit Parallelism

Slide 37

Holland believed that the GA with population size N searches more than N structures.

- A population of size N contains between 2^L and $N2^L$ schemata.
- i.e. it is sampling a large number of structures in parallel.
- How many are usefully sampled?

(argument not worth repeating; involves assumption of uniform distribution over schemata)

Slide 38

Answer: N^3 (Holland, Goldberg). "... with no special bookkeeping or memory other than the population itself."

Slide 39

What is wrong with Holland/Goldberg theory?

Finite population fluctuations: exponential increase conclusion is false.

Schemata are not independent: k-armed bandit analogy is false (Grefenstette 1989, 1993), e.g.

$$f(x) = \begin{cases} 2 \text{ if } x \in 111 * * * \cdots \\ 1 \text{ if } x \in 0 * * * * * \cdots \\ 0 \text{ otherwise.} \end{cases} \tag{10}$$

$u(1 * * * \cdots) = 1/2; u(0 * * * \cdots) = 1$ but the former is sampled more (as $111 * * * \cdots$).

Evidence against the building block hypothesis: at least as strong as the evidence for (e.g. Grefenstette 1993, Forrest and Mitchell 1993).

Slide 40

Markov Chain Approaches

Nix and Vose, 91; Vose and Liepins, 91; Vose, 96 Replace stochastic dynamics describing the states of the system at time t

With deterministic dynamics describing the probability of being in a given state at time t.

Process is Markovian. What are the states?

In a finite population, each population is a state.

$$\text{number of populations } = \binom{N + 2^L - 1}{2^L - 1} \approx \left(\frac{2^L}{N} e\right)^N. \tag{11}$$

Very complex structure to transition matrix. Very difficult to solve specific models.

Slide 41

Model — Royal Road Function

van Nimwegan et. al. 1997

Genotype: $\boxed{1\,1\,1\,0}\boxed{1\,1\,1\,1}\boxed{0\,1\,0\,1}\boxed{1\,1\,1\,1}\boxed{1\,1\,1\,1}$
M blocks of K bits each.

Fitness: Number of blocks of K 1's ($f = 3$ above).
Holland predicted that a GA would be very effective on this problem; Forrest and Mitchell found otherwise.

GA No crossover; fitness proportional selection and mutation.

Dynamics: Shows punctuated equilibrium; periods of stability interrupted by short periods of increasing fitness.

Slide 42

States: Describe in terms of fitness distribution

$$P(\vec{t}) = \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_M \end{pmatrix}$$

where P_i is the fraction of the population with fitness i.

Slide 43

Infinite population (no fluctuations)

$$\vec{P(t)} = \mathbf{MS}\vec{P(t)}$$

where $\mathbf{M}$ is the linear mutation operator and $\mathbf{S}$ is the non-linear (due to normalisation) selection operator.

Since normalisation is a linear operator, it can be done once at the end; resulting in a linear equation,

$$\vec{P(t)} \propto \mathbf{M\hat{S}}\vec{P(t)}.$$

The solution is

$$\vec{P(t)} \propto \left[\mathbf{M\hat{S}}\right]^t \vec{P(0)}$$

Evolves to the principal eigenvector of $\mathbf{M\hat{S}}$. All other eigenvectors are unstable fixed points.

Slide 44

Finite Population

Two changes

1. **Stochasticity**: $Pr[\vec{P} \to \vec{P}']$ is a multinomial with mean given by $\mathbf{MS}\vec{P}$.
2. $\vec{P}$ cannot change continuously; must change in increments of $1/N$. When

   $$E(\Delta P_f) = \left[\mathbf{MS}\vec{P}\right]_f - P_f \ll \frac{1}{N}$$

   a change in the population will be difficult to realize.

When is that?

Answer: When $\vec{P}$ is near an eigenvector of $\mathbf{MS}$.

- The kth level has k stable directions, $M - k$ unstable ones.

Slide 45

"Finite Populations Induce Metastability"

- Fitness of the levels is predicted by the model.
- Prediction of stasis time requires configurational information, as does the addition of crossover.

Slide 46

Macroscopic versus Microscopic descriptions

Microscopic description: describes the Genetic algorithm in terms of the individual strings in the population.

- Advantages — contains all of the information required to predict the next stage in the dynamics.
- Disadvantages — very complex. Contains much more information than we require.

Macroscopic description describe the population in terms of a small number of statistical quantities.

- Advantages — simple, potentially tractable, can focus on quantities of interest (e.g. mean fitness, expected best member, etc.)

Slide 47

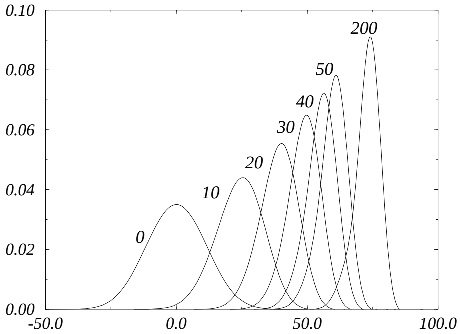
- Disadvantages — Much information is missing. Assumptions may be required to predict subsequent timesteps. missing information

Slide 48

Statistical Mechanics Approach

References: Prügel-Bennett, Rattray, Shapiro, Bornholdt.

- Predicts the dynamics of average statistical properties of the distribution of the objective function in the population (or other traits).



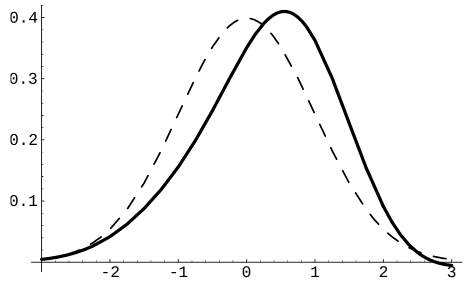
Slide 49

- Use Maximum Entropy inference to infer properties which are not simply related to the known statistics (e.g. configurational properties).

Slide 50

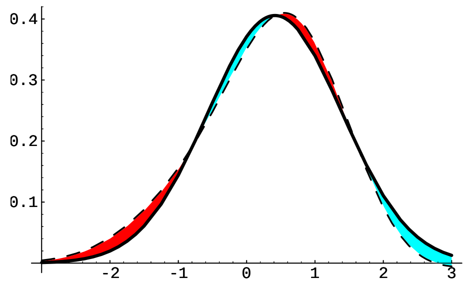
Seletion and Crossover Effects

Selection increases the mean, reduces the variance, and introduces skewness



Crossover restores higher cumulants such as skewness to random values, mediated by the degree of correlation in the population.

Slide 51



Slide 52

Optimising Parameters

These ideas can be use to implement a greedy search on parameters.

Selection: Selection increases the mean with little loss of diversity ($\sim 1/N$) up to a certain selection strength, beyond which diversity is lost without increase in the mean. Thus, selection pressure can be found which gives greatest increase in mean with least decrease in variance.

Mutation and Crossover: It has been suggested that it is possible to predict the expected best member of the population at each timestep, and chose the mutation rate and crossover probability to optimize that (Shapiro et. al. 1995). Bornholdt has presented results using this approach for mutation and selection (Bornholdt, 1999).

Slide 53

Questions

1. Can genetic algorithms learn effectively in complex environments?
2. How can open-ended evolution be assured in algorithms?
3. Can the evolutionary dynamics be understood, predicted, optimized?