

Genetic Algorithms

A class of probabilistic algorithms that are likely to find the global optimum of a multi-modal surface (no assumptions of differentiability-continuity)

They differ from purely random search technique in their exploitability of the information accumulated during the period of adaptation.

They find solutions to problems by an evolutionary process. Software design based on natural solution: a population of solutions is maintained. This population undergoes evolution in a form of natural selection: "good" solutions reproduce and "bad" solutions die, to be replaced by offspring of the good. An evaluation function (objective function) plays the role of the environment)

A Simple Genetic Algorithm

```

Procedure Genetic;
begin
  t := 0;
  Initialize P(t); {Population of solutions
                  x1t, ..., xNt N=population size}
  Evaluate P(t);   {Compute fitness of each solution}
  While not (termination-condition) do
  begin
    t := t+1;
    select P(t) from P(t-1);
    Recombine P(t);
    Evaluate P(t);
  end; {while}
end{Genetic}

```

Example

Maximize $f(x) = x^2 \quad x \in [0, 31]$

1. Choose a genetic representation of solutions to the problem; code the parameter x as a finite length binary string.
2. Start (Search) from a population of solutions, not a single solution chosen at random (ex. tossing of a coin)

x_1^0	01101	(13) ₁₀
x_2^0	11000	(24) ₁₀
x_3^0	01000	(8) ₁₀
x_4^0	10011	(19) ₁₀

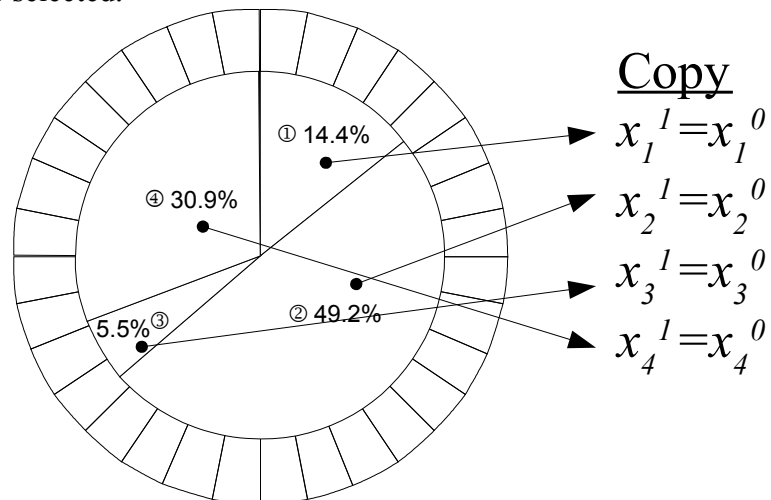
3. Evaluate/Rate solutions in terms of their fitness

$$f(x) \quad \% \text{fitness} = \frac{f(x'_i)}{\sum f(x'_i)}$$

x_1^0	169	14.4
x_2^0	576	49.2
x_3^0	64	5.5
x_4^0	361	30.9
Total	1170	100%
Avg	293	
Max	576	

4. {Natural selection} Select reproduction candidates using a biased roulette wheel (with slots sized according to fitness) i.e. higher fitness strings have more chance to be selected.

Above-average strings will have more offspring in the next generation



Apply genetic operations to form a new population (probabilistic transition rules).

Crossover

											$f(x)$
0	1	1	0	1	x_1'	01100	(12) ₁₀				144
1	1	0	0	0	x_2'	11001	(25) ₁₀				625
1	1	0	0	0	x_3'	11011	(27) ₁₀				729
1	0	0	1	1	x_4'	10000	(16) ₁₀				256
										Total	1754
										Avg	439
										Max	729

crossover = information exchange = testing of new parts of target regions

Analysis

A single string belongs to all the regions in which any of its bits appear

\therefore It belongs to 2^l schemata

\therefore A population of size N contains $2^l \rightarrow N \times 2^l$ schemata

ex

11011001 is a member of regions

11*****
1*****1
000*

A schema is a template describing a subset of strings

of schemata = 3^l

\therefore A genetic algorithm that manipulates a population of strings samples a large number of regions. (Implicit parallelism)

In the given examples consider three particular schemata

			$f(H)$	
$H_1 =$	1****	(x_2^0, x_4^0)	468.5	(x_2^1, x_3^1, x_4^1)
$H_2 =$	*10**	(x_2^0, x_3^0)	320	(x_2^1, x_3^1)
$H_3 =$	1***0	(x_2^0)	576	(x_4^1)

$$\text{on } \frac{f(H)}{f_{avg}} \text{ copies} = 2 \times \frac{468.5}{293} = 3.2$$

$$\delta(H_1) = 0$$

\therefore crossover has no effect

$$\frac{m \times \text{Pr}_m}{f_{avg}} \approx 3 \times 0.001 \approx 0.003$$

Properties of Schemata

The order of schema H : $O(H) = \#$ of fixed positions in the template

$$O(H), H = *10*0** = 3$$

Defining length of schema = $\delta(H)$ = distance between first and last fixed string positions.

$$\begin{aligned} \delta(H), H &= 011*1** = 4 \\ &= 0***** = 0 \end{aligned}$$

Highly fit short defining lengths schemata are called *building blocks*.

Effect of reproduction on the # of schemata

Suppose that within $P(t)$ there are m examples of a schema H. ($m = m(H, t)$)

During reproduction, a string x_2 is copied according to its fitness

$$\text{Pr}(x_1) = \frac{f_i}{\sum_{i=1}^N f_i} \quad (\text{population of size } N - \text{picking with replacement})$$

$$m(H, t+1) = m(H, t) \cdot \frac{f(H)}{\sum \frac{f_i}{N}} \rightarrow \begin{array}{l} \text{average fitness of strings matching schema H} \\ f_{avg} \end{array}$$

\therefore A particular schema grows as the ratio of the average fitness of the schema to the average fitness of the population.

Above-average schemata grow and below-average schemata die off.

Suppose H remains above average by a constant amount $c f_{avg}$

$$\therefore m(H, t+1) = m(H, t) \times \frac{f_{avg} + c f_{avg}}{f_{avg}} = m(H, t) \times (1 + c)$$

$$\therefore m(H, t) = m(H, 0) \times (1 + c)^t$$

\therefore Reproduction allocates exponentially increasing (decreasing) number of trials, above (below) average schemata.

Effect of crossover

A schema survives when the cross site falls outside its defining length

$$\text{Pr}_s = 1 - \frac{\delta(H)}{\ell - 1}$$

$$\begin{array}{l} \delta = 5 \\ \delta = 1 \end{array} \rightarrow \begin{array}{cccccc|cccc} * & 1 & * & & * & * & * & 0 & & & \\ * & * & * & & 1 & 0 & * & * & & & \end{array}$$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$
 $\ell - 1 = 6 = \text{possible sites for a cross}$

∴ Combined effect of reproduction & crossover

$$m(H, t+1) = \underbrace{m(H, t) \times \frac{f(H)}{f_{avg}}}_{\substack{\text{expected \# of schemata} \\ \text{due to reproduction} \\ \downarrow \\ \text{whether the schema is above or} \\ \text{below the population average}}} \times \underbrace{\left[1 - \frac{\delta(H)}{\ell-1} \right]}_{\substack{\text{survival probability} \\ \text{under crossover} \\ \downarrow \\ \text{Whether the schema has relatively} \\ \text{short or long defining length}}}$$

Mutation

Is the random alternation of a single position with probability

$$\Pr_m \approx 0.001$$

Survival probability of schema = $(1 - \Pr_m)^{O(H)}$

$$\approx 1 - O(H) \times \Pr_m$$

$$\therefore m(H, t+1) = m(H, t) \times \frac{f(H)}{f_{avg}} \left[1 - \frac{\delta(H)}{\ell-1} - O(H) \times \Pr_m \right]$$

∴ Low order short length above-average schemata receive exponentially increasing trials in subsequent generations. (schemata Theorem)