Basic GA theory

Fernando Lobo

University of Algarve

- Assume you have this population,

string	value
10111	10
01000	5
11010	3
00011	20

- Can you have to guess where other good solutions might be?

Basic GA theory

- Relies on the notion of schema.
- A schema is a similarity template that represents a set of solutions.
- For binary alphabets, a schema is a string over the symbols {0,1,*}
- H = 1**0* represents all strings that have a 1 in the 1st position and a 0 in the 4th position.
  - 10100 is a member (or instance) of H, but 00100 is not.

Order and defining length

- Two important definitions of a schema:
  - Order:
    - number of fixed positions.
  - Defining length:
    - distance of the two outermost fixed positions.

- Example: H = 1**0* has order 2 and defining length 3.

GAs as schema processors

- Holland recognized GAs as schema processors.
- While a GA evaluates a population of N strings, it implicitly evaluates a lot of different (much more than N) schemata.
- Example: Let 10100 have fitness 24.
  - The GA evaluates 10100 as 24.
  - But it also implicitly evaluates many partial solutions (1****, *0***, **1**, 10***, 1***0, …) as 24.

Holland's schema theorem

- Holland analysed the effects of the GA on an arbitrary schema H when going from one generation to the next.
- The analysis was done for proportionate selection and one-point crossover and resulted in the schema theorem.
- The theorem says that above average fitness schemata grow, and below average fitness schemata decay from generation to generation.

Holland's schema theorem

$$m(H,t+1) \;\geq\; m(H,t)\,\frac{\bar{f}(H,t)}{\bar{f}(t)}\left(1-p_c\,\frac{\delta(H)}{\ell-1}-p_m\;o(H)\right)$$

where:

$m(H,t)$ is the number of instances of schema H at time t,

$\bar{f}(H,t)$ is the average fitness of the instances of schema H at time t,

$\bar{f}(t)$ is the average fitness of the population at time t,

$\delta(H)$ is the defining length of schema H,

p_c is the probability of crossover,

p_m is the probability of mutation,

ℓ is the string length,

$o(H)$ is the order of schema H.

A more general schema theorem

- A more general schema theorem can be written as,

$$m(H, t+1) \geq m(H, t)\ \underbrace{\phi(H, t)}\ [1 - \underbrace{\epsilon(H, t)}]$$

Reproduction ratio
(due to selection)

Disruption factor
(due to variation ops)

- This version (due to Goldberg) is independent of the GA operators.

- Growth rate of a schema is: $\phi(H, t)\ [1 - \epsilon(H, t)]$

Building block hypothesis

- Holland's theory suggests that GAs work by combining highly fit sub-solutions (called building blocks).

- It does so by combining highly fit low order schemata into highly fit higher order schemata.

- Unfortunately, regular GAs can only process well a small fraction of those highly fit sub-solutions (those that have a short defining length, i.e. those whose linkage is tight).

Deception

- Consider a 5-bit problem where,

$$f(0****) > f(1****)$$
$$f(****0) > f(****1)$$

But,

$$f(0***0) < f(1***1)$$
$$f(0***1) < f(1***1)$$
$$f(1***0) < f(1***1)$$

Deception (cont.)

- The example illustrates a case where high-performance low-order schemata, 0**** and ****0, don't combine to form high-performance higher-order schemata.

- The GA is mislead to solutions belonging to 0***0, when it would be desirable to have solutions belonging to 1***1.
  - Having a 1 in the 1st position is not good by itself.
  - Having a 1 in the 5th position is also not good by itself.
  - But having a 1 simultaneously in the 1st and 5th position is good.

Deception (cont.)

- The previous problem is a 2-bit deceptive problem.
- Can be extended to k-bit deceptive.
- Those groups of k bits are usually called building blocks.
  - They have to be discovered at once
  - Crossover is unlikely to construct them by combining sub-parts of it.
  - As k increases the problem becomes more difficult (recall our lecture on problem difficulty).

Goldberg's decomposition for designing GAs

1. Know what GAs process: BBs.
2. Know BB challenging problems.
3. Ensure an adequate supply of BBs.
4. Ensure increased market share of BBs.
5. Understand takeover and convergence times.
6. Make decisions well among competing BBs.
7. Mix BBs well.

Goldberg's decomposition for designing GAs

- We have seen some of these issues (items 1, 2, 4).
- In the next lectures we will look at the remaning ones.