EAI 320
Practical 2

Edited and adapted: Michael Teles

Original: Hugo Coppejans and Johan Langenhoven

13 February 2020

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Today's focus

- Practical 1
  - Due 17 February 2020 at 23:59
  - Questions?
- Genetic Algorithms
  - Explanation
  - Example
- Practical 2
  - Check the online guide

Genetic Algorithm

- Genetic algorithm (GA) is a stochastic based searching algorithm.

- Good for optimisation as well – finding the best state out of a large number of different states.

- Sequence of events:
  1. Define fitness function – *f(n).*
  2. Evaluate each member of the *N*-sized population with *f(n)*.
  3. Choose best *n* from population. Discard *N-n* members.
  4. Repopulate by crossing over.
  5. Induce random mutation to keep population fresh.

- Some applications don't require mutations – depending on uncertainty of the data.

Genetic Algorithm (GA)

- Can get stuck in local minima/maxima.
- Mutation rates are difficult to refine and fine tune.
- Can take a very long time to find the solution, if the initial population is very far from the optimal state.
- Can potentially use some form of simulated annealing to adjust how the population mutates and regrows.
  - Choose initial mutation rate as very high.
  - As time goes by, decrease mutation rate.

Example 1

- This is an arbitrary example meant to give you a simple binary example of how crossover works. This example does not represent any particular problem and is entirely random.
- Given a set of 4 binary strings, use GAs to create three generations for the problem. Use the following rules while solving.
  - Each string has 8 genes (g1 – g8).
  - Population size of 4.
  - Use 50/50 crossover.
  - Fitness function is given as $f = 3g_1 - 2g_2 + 1g_3 - 4g_4 + 2g_5 + 2g_6 - 10g_7 + 5g_8$
  - Higher fitness is better.
  - Use the best two string to repopulate the population.
  - Thus the best two solutions stay, and the worst two gets replaced.
  - There is no mutation in this example.

Example 1

$$f = 3g_1 - 2g_2 + 1g_3 - 4g_4 + 2g_5 + 2g_6 - 10g_7 + 5g_8$$

First generation:

a = [1 1 0 1 0 1 1 1] ; f = -6
b = [0 0 1 0 0 1 0 1] ; f = 8
c = [1 1 0 0 1 0 0 0] ; f = 3
d = [1 1 0 1 1 1 1 0] ; f = -9

Second generation:

a = [0 0 1 0 1 0 0 0] ; f = 3
b = [0 0 1 0 0 1 0 1] ; f = 8
c = [1 1 0 0 1 0 0 0] ; f = 3
d = [1 1 0 0 0 1 0 1] ; f = 8

Example 1

$$f = 3g_1 - 2g_2 + 1g_3 - 4g_4 + 2g_5 + 2g_6 - 10g_7 + 5g_8$$

Second generation:

a = [0 0 1 0 1 0 0 0] ; f = 3
b = [0 0 1 0 0 1 0 1] ; f = 8
c = [1 1 0 0 1 0 0 0] ; f = 3
d = [1 1 0 0 0 1 0 1] ; f = 8

Third generation:

a = [0 0 1 0 0 1 0 1] ; f = 8
b = [0 0 1 0 0 1 0 1] ; f = 8
c = [1 1 0 0 0 1 0 1] ; f = 8
d = [1 1 0 0 0 1 0 1] ; f = 8

Example 1

- This example shows the main problem with GAs.

  If the gene pool becomes stagnant, the new chromosomes being created will have a similar fitness values to the parents. The solution to this is mutation and a larger population, or implementing mutations.

Practical 2

- 81 genes per string (‘R’, ‘P’ or ‘S’)
- History = [x_{t-2}, y_{t-2}, x_{t-1}, y_{t-1}]

History	RR	RP	RS	PR	PP	PS	SR	SP	SS
Output	P	S	R	R	S	R	R	S	R

Gene

String

Fig. 1 A reduced example only considering the history of the last move.

Practical 3

	RR	RP	RS	PR	PP	PS	SR	SP	SS
Parent 1	P	S	R	R	S	R	R	S	R
Parent 2	S	S	P	R	R	S	S	R	P
Reproduce	P	S	R	R	S	S	S	R	P
Mutate	P	R	R	R	S	S	S	R	P

Fig. 1 An example of the evolution step performed by the GA.

Practical 2

- Check the guide online.
- Preliminary due date - 24 February 2020 : 23h59.