Genetic Algorithms and Genetic Programming
Lecture 3

Gillian Hayes

29th September 2006

School of informatics

Lecture 3

Admin

- Tutorial groups: see email from ITO
- Preparation:
  - Read pp 1–27 of Mitchell
  - Read Whitley's GA tutorial sections 1 and 2 in lecture notes
- Lecture notes on paper???

Contents

Whitley's Canonical GA:

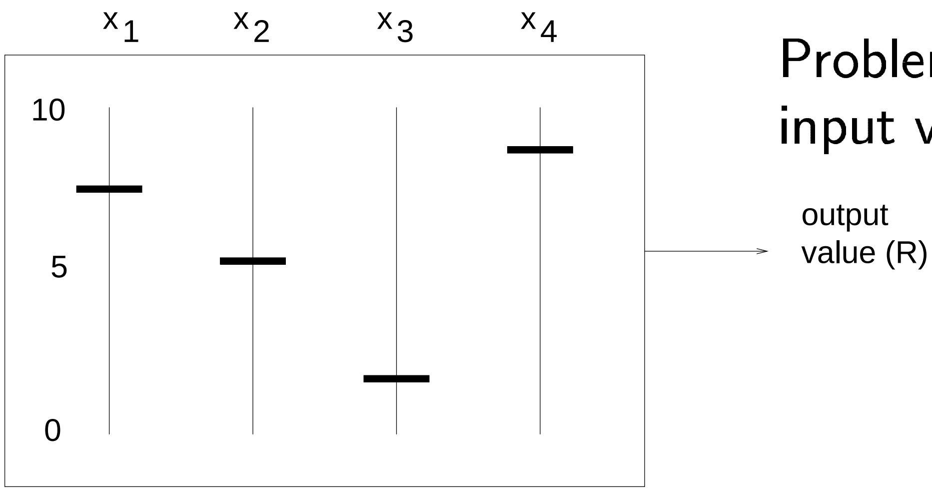
- representation
- evaluation/fitness
- selection
- crossover
- mutation
- an example

The Canonical GA

In the canonical GA, only 2 components are problem dependent:

- the problem encoding (representation)
- the evaluation function

Typical problem: parameter optimisation – find parameters that maximise R



Problem: output R is a nonlinear function of input values

$$R = F(x_1, x_2, x_3, x_4)$$

Interactions between parameters (epistasis) must be considered to maximise R.

Problem Encoding: Representation

In the canonical GA, solutions are encoded as binary integers (bit strings):

$$R = F(x_1, x_2, x_3, x_4)$$

x_1 : 0 to 31
x_2 : 0 to 1023
x_3 : 0 to 3
x_3 : 0 to 3
Total bits required $=$

If actual parameters are continuous then transform into a discrete range
$0 \rightarrow (2^n - 1)$ e.g. 0 to 1023

What if values are a finite set V where $|V| \neq 2^n$?

- map all 2^n values onto some value in V

- give impossible binary values a low evaluation

Evaluation Function

Evaluation function gives a score to each set of parameters:

$$f_i = F(x_1^i, x_2^i, x_3^i, \ldots)$$

for individual solution i.

If $\bar{f}$ is the average evaluation over the whole population of N individuals, then the **fitness** of i is $f_i/\bar{f}$ (so it's a relative fitness).

(Strictly, in the canonical GA, the fitness is this relative value. In GAs generally, you will find that "fitness" refers to f_i, $f_i/\bar{f}$, or some other function of f_i (e.g. to turn it into a cost rather than a fitness.)

Example:

$f_i = x_i^2, \;\; x = 00000, \ldots, 11111$ (i.e. 5-bit string, 0 to 31)

So f_i for $x_i = 01101$ (13) is 169.

If $\bar{f}$ for the whole population is 213.6 (say), then:

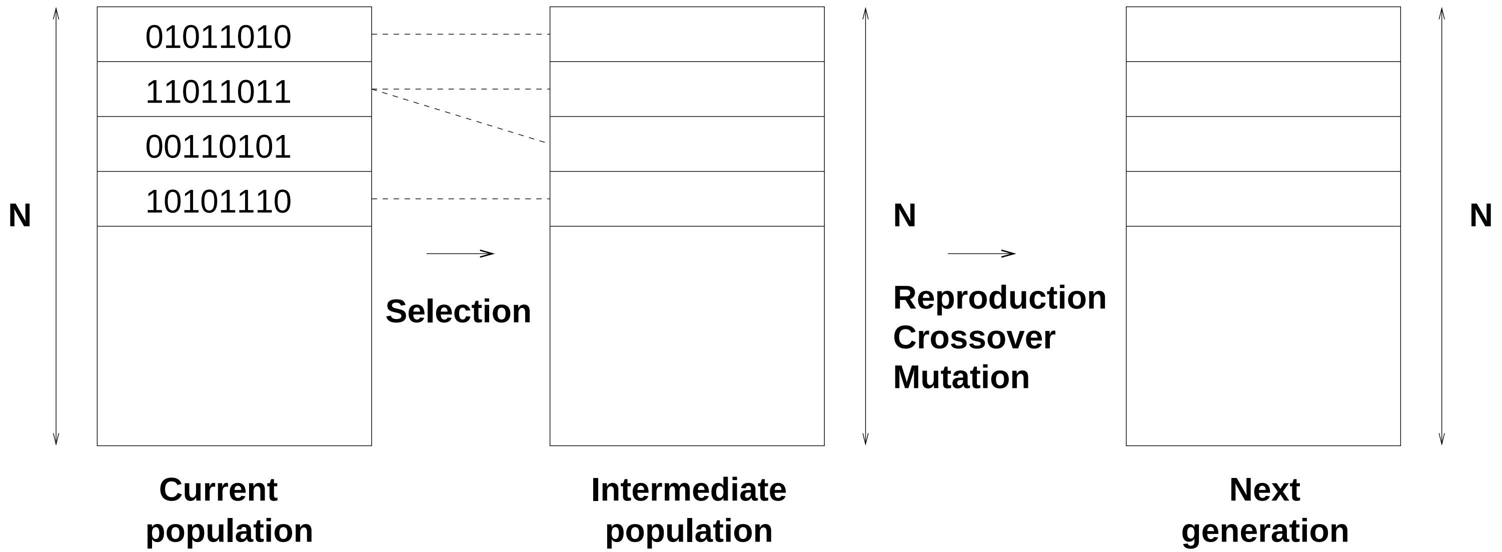
fitness(01101) $=$ 169/213.6 $=$ 0.79

What is the maximum evaluation f_i?

The maximum fitness $f_i/\bar{f}$?

Selection

In the canonical GA, selection maps the current population of size N onto an intermediate population of size N:



Roulette Wheel selection: each solution gets a chunk of the wheel which is proportional to its fitness (**fitness proportional selection**).

Spin the wheel N times to get N members of the **intermediate** population. Each time select chromosome with probability proportional to fitness – selection with replacement, so one chromosome can be selected many times. Known as **Stochastic sampling with replacement**.

Reproduction, Crossover and Mutation

The next generation is created by reproduction, crossover and mutation.

Select two parents at random from the intermediate population. Apply **crossover** with probability $= p_c$, with probability $= 1 - p_c$ copy the parents unchanged into the next generation – **reproduction**.

Crossover: from the 2 parents create 2 children using 1-point, 2-point, n-point crossover:

$$\begin{array}{lcl} \text{P1: } 01101|011011101 & & 01101101100100 \\ & \rightarrow & \\ \text{P2: } 11001|101100100 & & 11001011011101 \end{array}$$

Mutation: take each bit in turn and with Prob(mutation) $= p_m$, flip it ($0 \rightarrow 1$, $1 \rightarrow 0$). $p_m < 0.01$ usually.

This is one **generation**. Do for many generations, till solutions are optimal or good enough.

So:

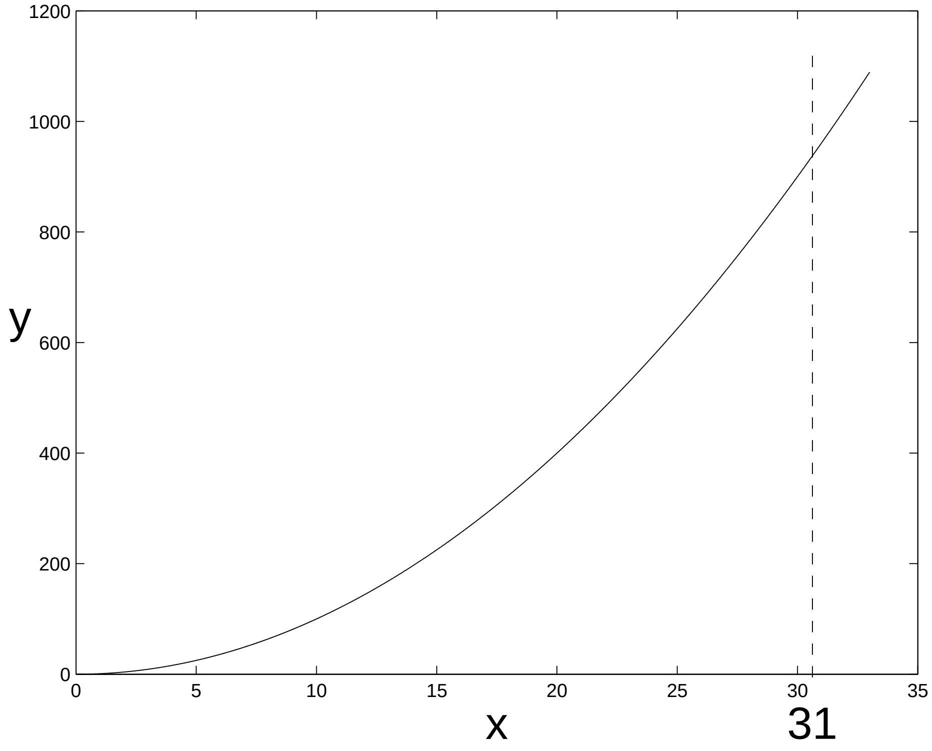
Repeat

- Evaluate fitness
- Select intermediate population
- Do crossover or reproduction
- Do mutation

Until solutions good enough

An Example

Maximise $y = x^2$ for x in the range 0 to 31. (What is the answer?)



Represent x as 5 bits:

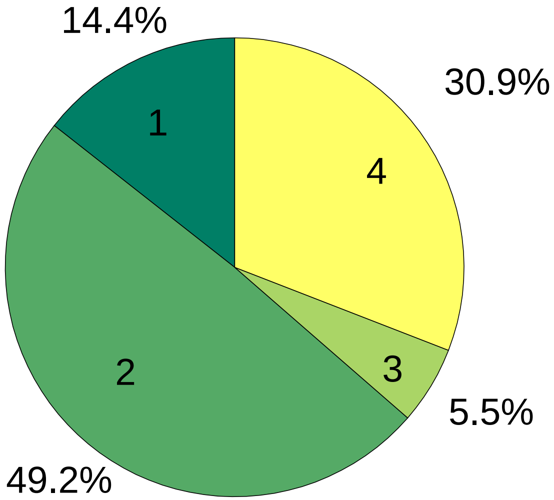
00000
00001
00010
$\vdots$
11111

Use a population of size 4 (far too small!)

Initial population:

i	Value	Evaluation	% of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9

Selection:



Suppose our intermediate population is: 1, 2, 2, 4

Create next generation:

$p_c = 1.0,\ p_r = 1 - p_c = 0.0$

Parents: 1 and 2

0110 1

$\rightarrow$

1100 0

Parents: 2 and 4

11 000

$\rightarrow$

10 011

Mutation: $p_m = 0.001$, 20 bits. No mutation.

New population:

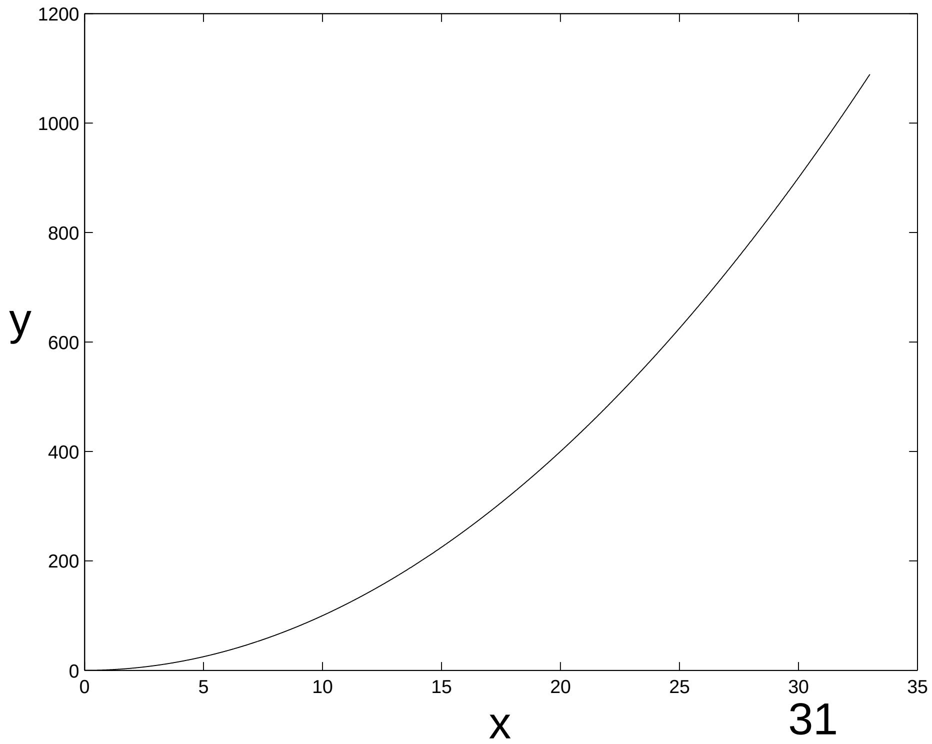
i	Value	Evaluation
1	01100	144
2	11001	625
3	11011	729
4	10000	256

What is the average evaluation of this population?

How does it compare with the average evaluation of the previous generation?

Continue until no improvement in the best solution for k generations, or run for fixed number of generations.

How does it work?



Climbing up the fitness curve.