

# **GENETIC ALGORITHMS: THE CONTINUOUS GA**

Date: Thursday 19 February 2015

Course: Functional Programming and Intelligent Algorithms

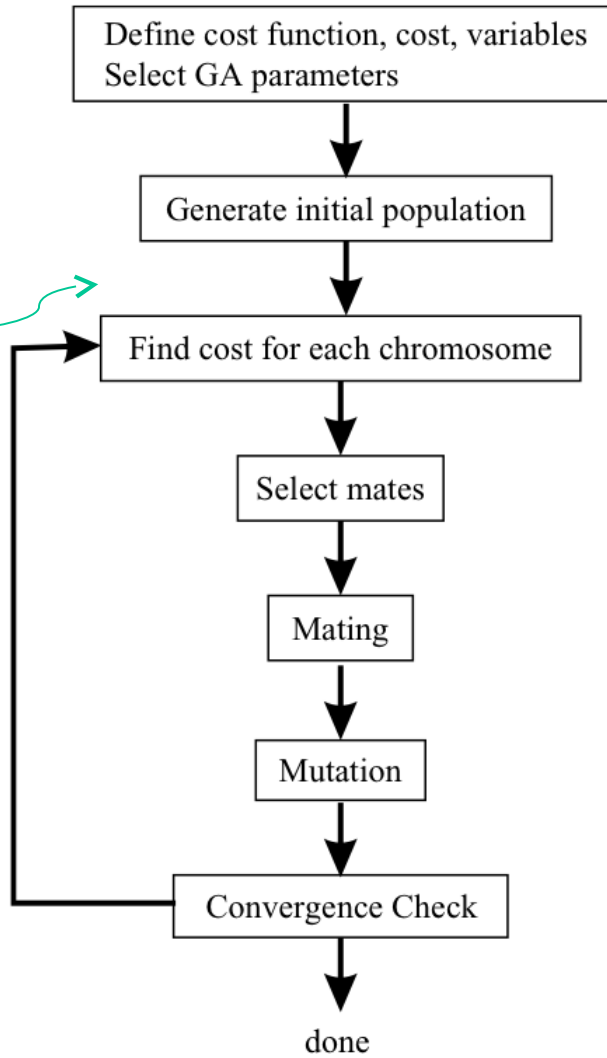
Lecturer: Robin T. Bye

# Components of the continuous GA

# Algorithm flow

1. Define cost function, cost, variables.  
Select GA parameters.
2. Generate initial population.
3. Find cost for each chromosome.
4. Select mates for reproduction.
5. Mating.
6. Mutation.
7. Convergence check (repeat from 3)

Compared with the binary GA, there is no step for decoding chromosomes here



**Figure 3.1** Flowchart of a continuous GA.

Adapted from [1].  
4

# Variables and cost function

- Nvar-dimensional problem  $\rightarrow$  chromosome has Nvar variables (genes),  $i=1, \dots, Nvar$
- $chrom = [p1, p2, \dots, pNvar]$
- $Cost = f(chrom) = f(p1, p2, \dots, pNvar)$
- Example: 2D height map in xy-plane
  - $chrom = [x, y]$
  - $cost = height = f(chrom) = f(x, y)$

# Variables and cost function

- No encoding to binary
- Use "continuous" values
  - but limited by computer precision, eg.,
    - floating points
    - double precision points

# Population

- Set of  $N_{pop}$  chromosomes
- Each chromosome is  $N_{var}$  row vector
- Represented as matrix of cts values
  - Dimensions are  $N_{pop} \times N_{var}$
- All variables  $p_i$  **normalised** to  $0 \leq p_i \leq 1$ 
  - "Unnormalise" in cost function
- Initial population randomly assigned:
  - `pop=rand(Npop, Nvar);`

# Unnormalisation

- Let
  - phi be the highest value of p
  - plo be the lowest value of p
  - pnorm be the normalised value of p
- Then the unnormalised value of p is
$$p = (\text{phi} - \text{plo})\text{pnorm} + \text{plo}$$



# Unnormalisation

- Example:
  - $\text{Phi} = 20$  and  $\text{plo} = -20$
  - $\text{pnorm} = 0.75$
- Then the unnormalised value of  $p$  is
$$\begin{aligned} p &= (\text{phi} - \text{plo})\text{pnorm} + \text{plo} \\ &= (20 - (-20)) * 0.75 + (-20) = 40 * 0.75 - 20 \\ &= 30 - 20 = 10 \end{aligned}$$

# Natural selection

1. Rank chromosomes (low cost better)
2. Only keep best fraction (=selection rate  $X_{rate}$ ) of  $N_{pop}$  chromosomes  $\Rightarrow$   
 $N_{keep} = X_{rate} \dagger N_{pop}$  chromosomes survives
3. Let kept chromosomes mate and replace discarded chromosomes

# Pairing methods

- From top to bottom (1+2, 3+4, etc.)
- Uniform random pairing
- Weighted random pairing
  - rank weighting
  - cost weighting
- Tournament selection
- Others

# Mating

- Many approaches
- Simple method: Just swap variable values at random crossover points
  - Problem: No new information introduced
  - Must rely on mutation for new genes
- Blending method (Radcliffe): Combine values of parents into new values
  - Blending limits values to interval of parents

# Mating

- Problem with point crossover methods:

$$parent_1 = [p_{m1}, p_{m2}, p_{m3}, p_{m4}, p_{m5}, p_{m6}, \dots, p_{mN_{var}}]$$

$$parent_2 = [p_{d1}, p_{d2}, p_{d3}, p_{d4}, p_{d5}, p_{d6}, \dots, p_{dN_{var}}]$$

$$offspring_1 = [p_{m1}, p_{m2}, \uparrow p_{d3}, p_{d4}, \uparrow p_{m5}, p_{m6}, \dots, p_{mN_{var}}]$$

$$offspring_2 = [p_{d1}, p_{d2}, \uparrow p_{m3}, p_{m4}, \uparrow p_{d5}, p_{d6}, \dots, p_{dN_{var}}]$$

- Merely interchanging data! No new genetic material generated!

# Mating

- Blending method:
  - $p_{new} = \beta p_{mn} + (1 - \beta) p_{dn}$
  - where
    - $\beta$  is random in range  $[0, 1]$
    - $p_{mn}$  is  $n$ th variable in mum chromosome
    - $p_{dn}$  is  $n$ th variable in dad chromosome
  - which variables to blend?
    - eg., all points to the right or left of crossover  $p$ .
    - only a selected few
  - can also use different  $\beta$  for each var

# Mating

- Blending method:
  - Works well on several interesting problems (Michalewicz, 1994)
  - Problem: Values are bracketed by the extremes already present in the population  $\Rightarrow$  must use an extrapolation method to avoid this.

# Mating

- Extrapolation method:
  - Extrapolation introduce values outside range of parents
  - Linear crossover:
    - Three offspring
    - $p_{new1} = 0.5p_{mn} + 0.5p_{dn}$  (average)
    - $p_{new2} = 1.5p_{mn} - 0.5p_{dn}$  (lower/higher)
    - $p_{new3} = -0.5p_{mn} + 1.5p_{dn}$  (higher/lower)
    - Variables outside bounds are discarded
    - Best two offspring are kept
    - Can use other factors than 0.5



# Mating

- Extrapolation method:
  - Example where  $p_{mn} = 4$ ,  $p_{dn} = 8$ 
    - $p_{new1} = 0.5p_{mn} + 0.5p_{dn} = 2 + 4 = 6$   
(average)
    - $p_{new2} = 1.5p_{mn} - 0.5p_{dn} = 6 - 4 = 2$  (lower)
    - $p_{new3} = -0.5p_{mn} + 1.5p_{dn} = -2 + 12 = 10$   
(higher)
  - We get a new lower, an average, and a new higher variable value

# Mating

- Extrapolation + crossover (Haupt):

Crossover point:  $\alpha = \mathbf{roundup}\{\mathbf{random} * N_{var}\}$   
– Randomly select a variable for crossover

Parents:

$$\begin{aligned} parent_1 &= [p_{m1} p_{m2} \dots p_{m\alpha} \dots p_{mN_{var}}] \\ parent_2 &= [p_{d1} p_{d2} \dots p_{d\alpha} \dots p_{dN_{var}}] \end{aligned}$$


Selected  
variables:

$$\begin{aligned} p_{new1} &= p_{m\alpha} - \beta[p_{m\alpha} - p_{d\alpha}] \\ p_{new2} &= p_{d\alpha} + \beta[p_{m\alpha} - p_{d\alpha}] \end{aligned}$$

# Mating

- Extrapolation + crossover (Haupt):

Offspring:

$$\text{offspring}_1 = [p_{m1} p_{m2} \dots p_{new1} \dots p_{dN_{var}}]$$
$$\text{offspring}_2 = [p_{d1} p_{d2} \dots p_{new2} \dots p_{mN_{var}}]$$


- All variables after crossover point are swapped in offspring
- $\beta$  is a random value in range  $[0,1]$ 
  - Limits offspring variables to bounds of parents
- Use  $\beta > 1$  for variables outside parent bounds
  - Must then perform a check for out-of-bounds

# Mutations

- Randomly mutate a fraction of values in list of chromosomes
  - New random (normalised) values in  $[0,1]$
- Can introduce novelty (new things)
  - Avoid early convergence to local minima
- Mutation rate  $\mu$  (mu), eg. 20%
- High mutation rate: Better exploration
  - But slower convergence

# Elitism

- Always keep best chromosome in population and never mutate it!
- Do not throw away a good solution!

# Next generation

- Insert offspring into population
- Recalculate costs and repeat process until
  - convergence
  - max number of iterations reached
  - you are happy for some reason

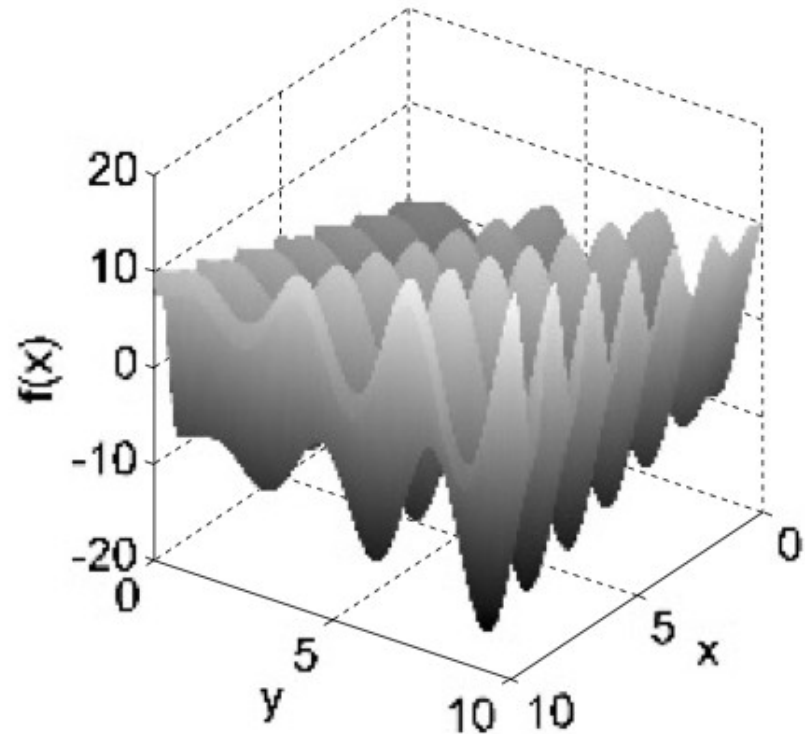
# Example 2D problem

Test function  $f_7(x,y)$

$$x \sin(4x) + 1.1y \sin(2y)$$

minimum:  $f(\overset{9.039, 8.668}{\cancel{0.9039, 0.8668}}) = -18.5547$

for  $0 \leq x, y \leq 10$



Adapted from [1].

# Example 2D problem

Initial population

Parents

**TABLE 3.1 Example Initial Population of 8 Random Chromosomes and Their Corresponding Cost**

$x$	$y$	Cost
6.9745	0.8342	3.4766
0.30359	9.6828	5.5408
2.402	9.3359	-2.2528
* 0.18758	8.9371	-8.0108
* 2.6974	6.2647	-2.8957
* 5.613	0.1289	-2.4601
* 7.7246	5.5655	-9.8884
6.8537	9.8784	13.752

**TABLE 3.2 Surviving Chromosomes after a 50% Selection Rate**

Number	$x$	$y$	Cost
1	7.7246	5.5655	-9.8884
2	0.1876	8.9371	-8.0108
3	2.6974	6.2647	-2.8957
4	5.6130	0.12885	-2.4601

\* best chromosomes

Adapted from [1].



# Example 2D problem

Pairs of random numbers to select mates:

(0.6710, 0.8124)  $\Rightarrow$  (2,3) from cumulative rank in table

(0.7930, 0.3039)  $\Rightarrow$  (3,1) from cumulative rank in table

**TABLE 2.5 Rank Weighting**

$n$	Chromosome	$P_n$	$\sum_{i=1}^n P_i$
1	00110010001100	0.4	0.4
2	11101100000001	0.3	0.7
3	00101111001000	0.2	0.9
4	00101111000110	0.1	1.0

# Example 2D problem

Create offspring and replace bad chromosomes

**TABLE 3.3 Pairing and Mating Process of Single-Point Crossover Chromosome Family Binary String Cost**

2	ma(1)	0.18758	8.9371
3	pa(1)	2.6974	6.2647
5	<i>offspring<sub>1</sub></i>	0.2558	6.2647
6	<i>offspring<sub>2</sub></i>	2.6292	8.9371
3	ma(2)	2.6974	6.2647
1	pa(2)	7.7246	5.5655
7	<i>offspring<sub>3</sub></i>	6.6676	5.5655
8	<i>offspring<sub>4</sub></i>	3.7544	6.2647

Mates (2,3)

Mates (3,1)

Adapted from [1].

# Example 2D problem

## 2nd and 3rd generations

**TABLE 3.6 Population after Crossover and Mutation in the Second Generation**

$x$	$y$	Cost
9.1602	5.5655	-14.05
2.6292	8.9371	-10.472
7.7246	6.4764	-1.1376
0.18758	8.9371	-8.0108
2.6292	5.8134	-7.496
9.1602	8.6892	-17.494
7.7246	8.6806	-13.339
4.4042	7.969	-6.1528

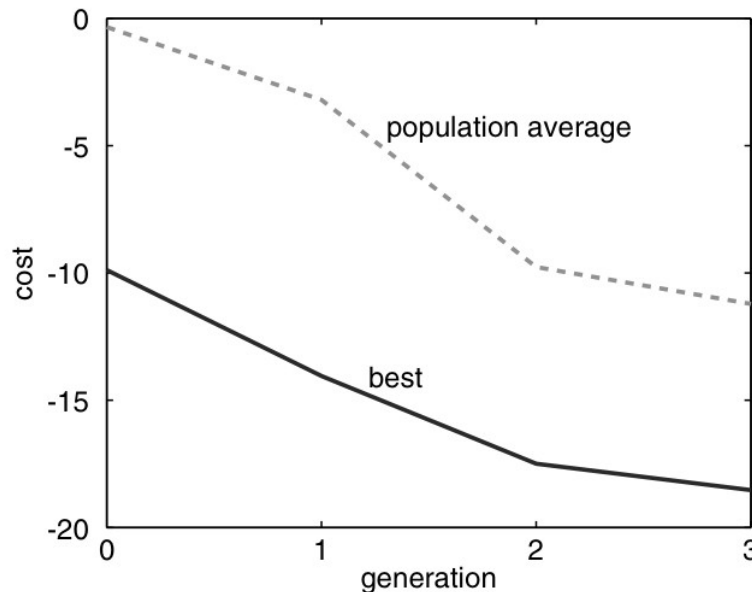
**TABLE 3.8 Ranking of Generation 3 from Least to Most Cost**

$x$	$y$	Cost
9.0215	8.6806	-18.53
9.1602	8.6892	-17.494
9.1602	8.323	-15.366
9.1602	5.5655	-14.05
9.1602	8.1917	-13.618
2.6292	8.9371	-10.472
7.7246	1.8372	-4.849
7.8633	3.995	4.6471

Adapted from [1].

# Example 2D problem

- Example converged after only 3 gen's
- $x_m=9.0215$ ,  $y_m=8.6806$ , cost = -18.53



**Figure 3.6** Plot of the minimum and mean costs as a function of generation. The algorithm converged in three generations.

# References

- [1] Haupt & Haupt, Practical Genetic Algorithms, 2nd Ed., Wiley, 2004.