Evolutionary Computation

Dirk Thierens

Utrecht University
The Netherlands

Course organization

- Part 1: lectures
- Part 2: practical assignment ⇒ report (groups of 2 students)
- Part 3: seminar ⇒ papers & presentation (student groups)

Course grading

1. Written exam = 60%
2. Practical assignment = 30%
3. Paper presentation = 10%

Pass = Total $\geq$ 6.0 and Minimum(Exam, Practical, Paper) $\geq$ 5.0

Qualify for the resit if Exam grade $\geq$ 4.0

Evolutionary Computation

= Population-based, stochastic search algorithms inspired by mechanisms of natural evolution

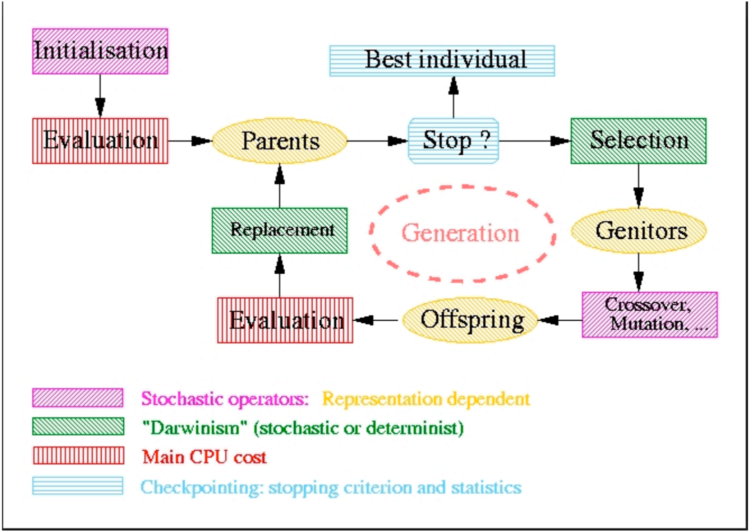
- EC part of Computational Intelligence
- Evolution viewed as search algorithm
- Natural evolution only used as metaphor for designing computational problem solving systems
- No modelling of natural evolution ($\neq$ evolutionary biology)

Key concepts of a Darwinian system

1. Information Structures
2. Copies
3. Variation
4. Competition
5. Inheritance

Evolutionary algorithm

1. `P(0) ← Generate-Random-Population()`
2. `P(0) ← Evaluate-Population(P(0))`
3. **while** `Not-Terminated?` **do**
    1. $P^s(t) \leftarrow$ `Select-Mates(P(t))`
    2. $P^o(t) \leftarrow$ `Generate-Offspring(`$P^s(t)$`)`
    3. $P^o(t) \leftarrow$ `Evaluate-Population(`$P^o(t)$`)`
    4. $P(t+1) \leftarrow$ `Select-Fittest(`$P^o(t) \cup P(t)$`)`
    5. $t \leftarrow t + 1$
4. **return** `P(t)`


Initialisation
Best individual
Evaluation
Parents
Stop ?
Selection
Replacement
Generation
Genitors
Evaluation
Offspring
Crossover, Mutation, ...
Stochastic operators: Representation dependent
"Darwinism" (stochastic or determinist)
Main CPU cost
Checkpointing: stopping criterion and statistics

Darwinian process characteristics: ⇒ Evolutionary Algorithm

1. **Information structures:**
   ⇒ e.g. binary strings, real-valued vectors, programs, ...
2. **Copies:**
   ⇒ selection algorithm
3. **Variation:**
   ⇒ mutation & crossover operators
4. **Competition:**
   ⇒ fitness based selection + fixed sized population
5. **Inheritance:**
   ⇒ Partial variation should lead to fitness correlation between parents and offspring

- **Neo-Darwinism**

organism

$\Uparrow$

...AUUCGCCAAU...

- **Genetic Algorithm**

f: $\Re$

$\Uparrow$

...0101001111...

* user: string representation and function **f**
* GA: string manipulation
  - **selection:** copy better strings
  - **variation:** generate new strings

Selection methods: fitness proportionate selection

- Probability P_i of selecting individual i with fitness value F_i

$$P_i = \frac{F_i}{\sum_{j=1}^{N} F_j} \qquad (N\text{: population size})$$

- Expected number of copies N_i of individual i

$$N_i = N \times P_i = \frac{F_i}{\bar{F}} \qquad (\bar{F}\text{: population mean fitness})$$

- Number of individuals with above average fitness increases
- Problems:
  1. Too much selection pressure if single individual has much higher fitness than the others in the population
  2. Loss of selection pressure when all fitness values converge to similar values

Selection methods: ranked based

Selection based on relative fitness as opposed to absolute fitness

1. Truncation selection
   - Sort the population according to the fitness values
   - Select the top $\tau\%$
   - Copy each selected individual $\frac{100}{\tau}$ times
2. Tournament selection
   - Select best individual from K randomly selected individuals (preferably selected without replacement)
   - Hold N tournaments to select N parent solutions

Selection pressure can be tuned by changing the truncation threshold τ or the tournament size K

Variation methods: mutation & crossover

1. mutation

$$\{1111111111 \Rightarrow \{1111111011$$

(small perturbations should be more likely than large ones)

2. crossover

$$\text{2-point crossover:} \begin{cases} 1111111111 \\ 0000000000 \end{cases} \Rightarrow \begin{cases} 1111000011 \\ 0000111100 \end{cases}$$

$$\text{uniform crossover:} \begin{cases} 1111111111 \\ 0000000000 \end{cases} \Rightarrow \begin{cases} 1001110101 \\ 0110001010 \end{cases}$$

Toy example

$$x \epsilon [0, 31] \; : \; f(x) = x^2$$

binary integer representation: $x_i \epsilon \{0, 1\}$

$$x = x_1 * 2^4 + x_2 * 2^3 + x_3 * 2^2 + x_4 * 2^1 + x_5 * 2^0$$

- Initial Random Population:

$$\begin{aligned} 10010 &: 18^2 = 324 \\ 01100 &: 12^2 = 144 \\ 01001 &: 9^2 = 81 \\ 10100 &: 20^2 = 400 \\ 01000 &: 8^2 = 64 \\ 00111 &: 7^2 = 49 \end{aligned}$$

population mean fitness $\bar{f}(0) = 177$

- Generation 1:
  tournament selection, 1-point crossover, mutation

Parents	Fitness	Offspring	Fitness
100|10	324	10100	400
101|00	400	10111	529
01|000	64	00010	4
10|010	324	10010	324
0110|0	144	11100	784
1010|0	400	10000	256

Parent population mean fitness $\bar{f}(1) = 383$

- Generation 3:

Parents	Fitness	Offspring	Fitness
1|1111	961	11110	900
1|1100	784	11011	729
110|00	576	11110	900
111|10	900	11101	841
1101|1	729	11111	961
1100|1	625	01001	81

Parent population mean fitness $\bar{f}(3) = 762$

Schemata

- Schema = similarity subset

$$11\#\#0 = \{11000, 11010, 11100, 11110\}$$

- How does the number of solutions that are member of particular schemata change in successive populations ?

generation	1#####	0#####	#####1	#####0
0	2	4	2	4
1	5	1	1	5
2	6	0	2	4
3	6	0	3	3
4	6	0	3	3
5	5	1	4	2

Schemata definitions

- $o(h)$: schema order = number of fixed values: $o(11\#\#0) = 3$
- δ(h): schema length = distance between leftmost and rightmost fixed position: $\delta(\#11\#\#0) = 4$
- $m(h, t)$: number of schema h instances at generation t
- $f(h, t) = \overline{\sum_{i \in P} f_i}$: schema fitness is average fitness of individual members

Schemata competition

- key issue: **changing number of schemata members in successive population**.
- fit schemata increase in proportion by selection.
- Schemata compete in their respective partitioning:

$$\#\#f\#f : \#\#0\#0, \#\#0\#1, \#\#1\#0, \#\#1\#1$$

- Mutation and crossover viewed as destructive operators for the fit schemata.

Schema growth by selection

- Reproduction ratio $\phi(h,t)$

$$\boxed{\phi(h,t) = \frac{m(h,t^s)}{m(h,t)}}$$

- **proportionate selection**
  - Probability individual i selected: $\frac{f_i}{\sum f_i}$ (f_i: fitness ind. i)
  - Expected number of copies of ind. i : $\frac{f_i}{\sum f_i}.N = \frac{f_i}{\bar{f}(t)}$ (N: population size)
  - Expected number of copies of schema h members:

$$m(h,t^s) = m(h,t)\phi(h,t) = m(h,t)\frac{f(h,t)}{\bar{f}(t)}$$

- **tournament selection**
  - tournament size K: $0 \leq \phi(h,t) \leq K$

Schema disruption by mutation

- probability bit flipped: p_m
- schema h survives iff all the bit values are *not* mutated

$$p_{survival} = (1 - p_m)^{o(h)}$$

- for small values $p_m << 1$

$$(1 - p_m)^{o(h)} \approx 1 - o(h).p_m$$

- disruption factor $\epsilon(h, t)$ by mutation:

$$\boxed{\epsilon(h, t) = o(h).p_m}$$

Schema disruption by recombination

- probability crossover applied p_c
- **1-point crossover**
  - schema h survives iff cutpoint *not* within schema length δ:

$$p_{survival} = 1 - \frac{\delta(h,t)}{l-1}$$

- **uniform crossover** (bit swap probability: p_x)
  - schema h survives iff none or all bits swapped together

$$p_{survival} = p_x^{o(h)} + (1-p_x)^{o(h)}$$

- disruption factor $\epsilon(h,t)$ by recombination:

$$\boxed{\epsilon(h,t) = p_c.(1 - p_{survival})}$$

(p_c: probability of applying crossover)

Schema Theorem

- Selection, mutation, and recombination combined:

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

- net growth factor: $\gamma(h, t) = \frac{m(h,t+1)}{m(h,t)}$

$$\boxed{\gamma(h, t) \geq \phi(h, t)[1 - \epsilon(h, t)]}$$

schemata with $\gamma(h, t) > 1$ increase in proportion
schemata with $\gamma(h, t) < 1$ decrease in proportion

Schema Theorem cont'd

- low order, high performance schemata receive exponentially (geometrically) increasing trials → **building blocks**
- according to the k-armed bandit analogy this strategy is near optimal (Holland, 1975)
- happens in an implicit parallel way

  → only the short, low-order schemata are processed reliably

- enough samples present for statistically reliable information
- enough samples survive the disruption of variation operators

Building Blocks

Building block hypothesis

= building blocks can be juxtaposed to form near optimal solutions

Consequences

1. schema sampling is a statistical decision process: **variance considerations**
2. building blocks must be juxtaposed before convergence: **mixing analysis**
3. low order schemata might give misleading information: **deceptive problems**