7. Genetic Algorithms

7.1 Introduction
Genetic Algorithms: Part 1

Outline

- Evolutionary Computing (EC)
- Biological Background
- Landscape Example
- Natural Genetics
- What is a Genetic Algorithm?
- Simple Genetic Algorithm (SGA)
- References
Evolutionary Computing (EC)
Evolutionary Computing (EC)

- EC is part of computer science
- EC is not part of life sciences/biology
- It draws inspiration from the process of natural evolution
- EC can be applied in biological research
### The Main EC Metaphor

<table>
<thead>
<tr>
<th>EC</th>
<th>Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization problem</td>
<td>Environment</td>
</tr>
<tr>
<td>Feasible solutions</td>
<td>Individuals living in that environment</td>
</tr>
<tr>
<td>Solutions quality</td>
<td>Fitness (Individual’s degree of adaptation to its surrounding environment)</td>
</tr>
</tbody>
</table>
Evolutionary Computing Areas

- Genetic Programming
- Evolution Strategies
- Genetic Algorithms
- Evolutionary Programming
Brief History

• 1964, Rechenberg introduces evolution strategies
• 1965, L. Fogel, Owens and Walsh introduce evolutionary programming
• 1975, Holland introduces genetic algorithms
• 1992, Koza introduces genetic programming
Motivations for EC

- Nature has always served as a source of inspiration for engineers and scientists.
- The best problem solver known in nature is:
  - the (human) brain that created “the wheel, New York, wars and so on”
  - the evolution mechanism that created the human brain
Motivations for EC

- Developing, analyzing, applying problem solving methods (algorithms) is a central theme in mathematics and computer science.
- **Complexity** of problems to be solved increases.
- Consequence: **Robust problem solving** technology needed
  - Which do not need much tailoring for specific problems, and
  - Deliver good (not necessarily optimal) solutions within acceptable time.
- EC do all this.
Biological Background
Darwin’s principles

- **Variety** of species individuals within the population
- **Overproduction** of offspring generation
  - Individuals have basic instinct towards reproduction
- **Competition** for limited resources
  - Environment only support a limited number of individuals
- **Survival of the fittest**
  - Those individuals, that are adopted or fit to the environmental condition best, have increased chance of reproduction
Evolution

How does it work?

- Initial population
  - Variety of shapes, colors, behaviors
  - Each individual fits differently to the environment
Genetic Algorithms: Part 1

Evolution

How does it work?

- Initial population
- Reproduction
  - Offspring combines both parents properties
  - Siblings may differ in properties
  - Mutations may occur
Evolution

How does it work?

- Initial population
- Reproduction
- Limited environmental resources
  - Only a portion of the individuals survive
  - Survival chances – according to fitness measure
Genetic Algorithms: Part 1

Evolution

- **Phenotypic traits**:  
  - Behaviour / physical differences that affect **response to environment**  
  - Partly determined by **inheritance**, partly by factors **during development**  
  - Unique to each individual, partly as a result of **random changes**

- If phenotypic traits lead to higher chances of reproduction, then  
  - Can be inherited  
  - They will tend to increase in subsequent generations  
  - Leading to new combinations of traits …
Natural Genetics
Natural Genetics

- The information required to build a living organism is coded in the DNA of that organism.
- **Genotype** (DNA inside) determines **Phenotype**.
Natural Genetics

- **Genes** are encoding phenotypic characteristics
  - One gene may affect many traits
  - Many genes may affect one trait
- Small changes in the genotype lead to small changes in the organism (e.g., height, hair colour)
- The possibilities of the **genes** for one property is called **Allele**
- Genotypic variations are consequences of:
  - **Recombination** of genes by sexual reproduction
  - **Mutation** of genes
Genes and the Genome

- The complete genetic information in an individual’s genotype is called the **Genome**
- Genes are encoded in strings of DNA called **Chromosomes**
- In most **cells**, there are two copies of each chromosome, called **Diploid**
- Within a species, most of the genetic material is the same
Example

- **Human body cells** contains 23 pairs of chromosomes which together define the attributes of the individual:
Reproductive Cells

- **Gametes** (i.e., sperm and egg cells) contain 23 individual chromosomes rather than 23 pairs.
- Cells (gametes) with only one copy of each chromosome are called **Haploid**.
- The haploid sperm cell merges with the haploid egg cell and forms a diploid cell, called **Zygote**.
- The new organism develops from this zygote by the process named **Ontogenesis**.
- All body cells contain the **same** genetic information as the zygote in its original form.
Mitosis

- **Mitosis** is copying the same genetic information to new offspring.
- Mitosis is the normal way of growing multicell structures.
Meiosis

- **Meiosis** is the basis of sexual reproduction.
- After meiotic division, gametes appear in the process.
- Hence genetic information is shared between the parents in order to create new offspring.
- During meiosis the pairs of chromosome undergo an operation called **Crossing over**.
Genetic Algorithms: Part 1

Crossing-over during meiosis

- Chromosome pairs align and duplicate
- Inner pairs link exchange parts of themselves

- Outcome is one copy of maternal/paternal chromosome plus two entirely new combinations
- After crossing over one of each pair goes into each gamete
Fertilisation

Sperm cell from Father

Egg cell from Mother

New person cell (zygote)
After fertilisation

- **New zygote** rapidly divides and creating many cells all with the same genetic contents.

- Although all cells contain the same genes, depending on, for example where they are in the organism, they will behave differently.

- This process of differential behaviour during development is called **ontogenesis**.

- All of this uses, and is controlled by, the same mechanism for decoding the genes in DNA.
A central claim in molecular genetics: only one way flow

Genotype $\rightarrow$ Phenotype

Genotype $\leftarrow$ Phenotype

**Lamarckism** (saying that acquired features can be inherited) is thus wrong!
Mutation

- Occasionally some of the genetic material changes very slightly during this process.
- This means that the child might have genetic material information **not inherited** from either parent.
- This can be:
  - **Disastrous**: offspring in not viable (most likely)
  - **Neutral**: new feature not influences fitness
  - **Advantageous**: strong new feature occurs
Landscape Example
Landscape metaphor

- The **height dimension** belongs to fitness
- The other two (or more) dimensions correspond to **biological traits**
- The x-y-plane holds all possible trait combinations
- Therefore, each different individual (phenotype) represents a single point on the landscape
- Population is therefore a “cloud” of points, moving on the landscape over time as it evolves - adaptation
Genetic Algorithms: Part 1

Example with two traits
Genetic Algorithms: Part 1

Landscape metaphor

- Selection “pushes” population up the landscape
- There are a number of points that are better than all their neighbouring solutions, we call each of these points a **local optimum**
- The highest of these points is called **global optimum**
- Random variations in feature distribution (+ or -) arising from sampling error can cause the population down hills, thus crossing valleys and leaving local optima
What is a Genetic Algorithm?
A population of individuals exists in an environment with limited resources.

**Competition** for those resources causes selection of those fitter individuals that are better adapted to the environment.

These individuals act as seeds for the generation of new individuals through recombination and mutation.

The new individuals have their fitness evaluated and compete for survival.

Over time **natural selection** causes a rise in the fitness of the population.
Genetic Algorithms: Part 1

Definition

- **Genetic Algorithms** are
  - Bio-Inspired artificial intelligence class,
  - stochastic,
  - *population-based* algorithms

- Typically applied to:
  - hard problems with a large search space
  - discrete optimization

- Developed by **John Holland**, USA in the 1970’s
BEGIN

INITIALISE population with random candidate solutions;
EVALUATE each candidate;
REPEAT UNTIL (TERMINATION CONDITION is satisfied) DO
  1 SELECT parents;
  2 RECOMBINE pairs of parents;
  3 MUTATE the resulting offspring;
  4 EVALUATE new candidates;
  5 SELECT individuals for the next generation;
OD
END
Genetic Algorithms: Part 1

GA Algorithmic Phases

- Initialize the population
- Select Parents
- Recombination
- Mutation
- Select Individuals for Next Generation
- Stop?
  - yes: The End
  - no: Repeat the process
Genetic Algorithms: Part 1

General Scheme of GA

- Initialisation
- Population
  - Parent selection
  - Recombination
  - Mutation
- Offspring
  - Survivor selection
- Termination
Simple Genetic Algorithm (SGA)
Simple Genetic Algorithm (SGA)

- Holland’s original GA is now known as the simple genetic algorithm (SGA)

- Other GAs use different:
  - Representations
  - Mutations
  - Crossovers
  - Selection mechanisms
### SGA summary

<table>
<thead>
<tr>
<th>Representation</th>
<th>Binary strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recombination</td>
<td>1-point crossover</td>
</tr>
<tr>
<td>Mutation</td>
<td>bit-flipping with fixed probability</td>
</tr>
<tr>
<td>Parent selection</td>
<td>Fitness-Proportionate</td>
</tr>
<tr>
<td>Survivor selection</td>
<td>All children replace parents</td>
</tr>
<tr>
<td>Speciality</td>
<td>Emphasis on crossover</td>
</tr>
</tbody>
</table>
Genetic Algorithms: Part 1

Simple example – \( f(x) = x^2 \)

- Finding the maximum of a function:
  - \( f(x) = x^2 \)
  - Range \([0, 31]\) → Goal: find max \((31^2 = 961)\)
  - Binary representation: string length 5 = 32 numbers (0-31)

<table>
<thead>
<tr>
<th>genotype</th>
<th>0 0 1 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapping</td>
<td>2 2 2 2 2</td>
</tr>
<tr>
<td></td>
<td>16 8 4 2 1</td>
</tr>
<tr>
<td>phenotype</td>
<td>0<em>16+0</em>8+1<em>4+0</em>2+1*1 = 5</td>
</tr>
<tr>
<td>fitness</td>
<td>25 = f(x)</td>
</tr>
</tbody>
</table>
Genetic Algorithms: Part 1

\(x^2\) example

- **\(x^2\) example:**
  - Representation: Binary code
  - Population size: 4
  - Recombination: 1-point crossover
  - Mutation: Bit-flipping with fixed probability
  - Parent selection: Fitness-Proportionate
  - Initialization: Random

- We show one generational cycle done by hand
### Genetic Algorithms: Part 1

**x² example: selection**

<table>
<thead>
<tr>
<th>String no.</th>
<th>Initial population</th>
<th>$x$ Value</th>
<th>Fitness $f(x) = x^2$</th>
<th>$Prob_i$</th>
<th>Expected count</th>
<th>Actual count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 1 1 0 1</td>
<td>13</td>
<td>169</td>
<td>0.14</td>
<td>0.58</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1 1 0 0 0</td>
<td>24</td>
<td>576</td>
<td>0.49</td>
<td>1.97</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 0 0</td>
<td>8</td>
<td>64</td>
<td>0.06</td>
<td>0.22</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0 1 1</td>
<td>19</td>
<td>361</td>
<td>0.31</td>
<td>1.23</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td>4.00</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td>0.25</td>
<td>1.00</td>
<td>1</td>
</tr>
<tr>
<td>Max</td>
<td></td>
<td></td>
<td></td>
<td>0.49</td>
<td>1.97</td>
<td>2</td>
</tr>
</tbody>
</table>
### X² example: crossover

<table>
<thead>
<tr>
<th>String no.</th>
<th>Mating pool</th>
<th>Crossover point</th>
<th>Offspring after xover</th>
<th>x Value</th>
<th>Fitness ( f(x) = x^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 1 1 0</td>
<td>1</td>
<td>0 1 1 0 0</td>
<td>12</td>
<td>144</td>
</tr>
<tr>
<td>2</td>
<td>1 1 0 0</td>
<td>0</td>
<td>1 1 0 0 1</td>
<td>25</td>
<td>625</td>
</tr>
<tr>
<td>2</td>
<td>1 1</td>
<td>0 0 0</td>
<td>1 1 0 1 1</td>
<td>27</td>
<td>729</td>
</tr>
<tr>
<td>4</td>
<td>1 0</td>
<td>0 1 1</td>
<td>1 0 0 0 0</td>
<td>16</td>
<td>256</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1754</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>439</td>
</tr>
<tr>
<td>Max</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>729</td>
</tr>
</tbody>
</table>
### X² example: mutation

<table>
<thead>
<tr>
<th>String no.</th>
<th>Offspring after xover</th>
<th>Offspring after mutation</th>
<th>$x$ Value</th>
<th>Fitness $f(x) = x^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01100</td>
<td>11100</td>
<td>26</td>
<td>676</td>
</tr>
<tr>
<td>2</td>
<td>11001</td>
<td>11001</td>
<td>25</td>
<td>625</td>
</tr>
<tr>
<td>2</td>
<td>11011</td>
<td>11011</td>
<td>27</td>
<td>729</td>
</tr>
<tr>
<td>4</td>
<td>10000</td>
<td>10100</td>
<td>18</td>
<td>324</td>
</tr>
<tr>
<td>Sum Average Max</td>
<td></td>
<td></td>
<td></td>
<td>2354 588.5 729</td>
</tr>
</tbody>
</table>
SGA Shows many shortcomings:
- Representation is too restrictive
- Mutation & crossovers only applicable for bit-string & integer representations
- Selection mechanism sensitive for converging populations with close fitness values
- Generational population model (step 5 in SGA) can be improved with explicit survivor selection
References
References

The End