Contents

- Introduction
- Generic Evolutionary Algorithm
- Representation
- Initial Population
- Fitness Function
- Selection
- Reproduction Operators
- Stopping Conditions
- Evolutionary Computation versus Classical Optimization
Some Theories about Evolution

- **Evolution is an optimization process:**
  - the aim is to improve the ability of an organism to **survive** in dynamically changing and competitive environments

- **Two main theories on biological evolution:**
  - Lamarckian (1744-1829) view
  - Darwinian (1809-1882) view
Lamarckian Evolution

- *Heredity*, i.e. the inheritance of acquired traits
- Individuals adapt during their lifetimes, and transmit their traits to their offspring
- Offspring continue to adapt
- Method of adaptation rests on the concept of use and disuse
- Over time individuals lose characteristics they do not require and develop those which are useful by “exercising” them
Darwinian Evolution

- Theory of **natural selection** and **survival of the fittest**
- In a world with limited resources and stable populations, each individual competes with others for survival
- Those individuals with the “best” characteristics (traits) are more likely to survive and to reproduce, and those characteristics will be passed on to their offspring
- These desirable characteristics are inherited by the following generations, and (over time) become dominant among the population
- During production of a child organism, random events cause random changes to the child organism’s characteristics
- What about Alfred Wallace?
Evolutionary Computation

Evolutionary computation (EC) refers to computer-based problem solving systems that use computational models of evolutionary processes, such as

- natural selection,
- survival of the fittest, and
- reproduction,

as the fundamental components of such computational systems.
Main Components of an Evolutionary Algorithm

- Evolution via natural selection of a randomly chosen population of individuals can be thought of as a search through the space of possible chromosome values.
- In this sense, an evolutionary algorithm (EA) is a stochastic search for an optimal solution to a given problem.
- Main components of an EA:
  - an encoding of solutions to the problem as a chromosome;
  - a function to evaluate the fitness, or survival strength of individuals;
  - initialization of the initial population;
  - selection operators; and
  - reproduction operators.
Generic Evolutionary Algorithm: Algorithm 8.1

Let \( t = 0 \) be the generation counter;
Create and initialize an \( n_x \)-dimensional population, \( C(0) \), to consist of \( n_s \) individuals;
\[ \textbf{while stopping condition(s) not true do} \]
\[ \text{Evaluate the fitness, } f(x_i(t)), \text{ of each individual, } x_i(t); \]
\[ \text{Perform reproduction to create offspring;} \]
\[ \text{Select the new population, } C(t + 1); \]
\[ \text{Advance to the new generation, i.e. } t = t + 1; \]
\[ \textbf{end} \]
Evolutionary Computation Paradigms

- **Genetic algorithms** model genetic evolution.
- **Genetic programming**, based on genetic algorithms, but individuals are programs.
- **Evolutionary programming**, derived from the simulation of adaptive behavior in evolution (i.e. *phenotypic* evolution).
- **Evolution strategies**, geared toward modeling the strategic parameters that control variation in evolution, i.e. the evolution of evolution.
- **Differential evolution**, similar to genetic algorithms, differing in the reproduction mechanism used.
- **Cultural evolution**, which models the evolution of culture.
- **Co-evolution**, in competition or cooperation.
The Chromosome

- Characteristics of individuals are represented by long strings of information contained in the *chromosomes* of the organism.
- Chromosomes are structures of compact intertwined molecules of DNA, found in the nucleus of organic cells.
- Each chromosome contains a large number of genes, where a *gene* is the unit of heredity.
- An alternative form of a gene is referred to as an *allele*. 
The “Artificial” Chromosome

In the context of EC, the following notation is used:

<table>
<thead>
<tr>
<th>Chromosome (genome)</th>
<th>Candidate solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene</td>
<td>A single variable to be optimized</td>
</tr>
<tr>
<td>Allele</td>
<td>Assignment of a value to a variable</td>
</tr>
</tbody>
</table>

Two classes of evolutionary information:

- A *genotype* describes the genetic composition of an individual, as inherited from its parents.
- A *phenotype* is the expressed behavioral traits of an individual in a specific environment.
Binary Representation

- Binary-valued variables: $x_i \in \{0, 1\}^n$, i.e. each $x_{ij} \in \{0, 1\}$
- Nominal-valued variables: each nominal value can be encoded as an $n_d$-dimensional bit vector where $2^{n_d}$ is the total number of discrete nominal values for that variable
- Continuous-valued variables: map the continuous search space to a binary-valued search space:
  - Each continuous-valued variable is mapped to an $n_d$-dimensional bit vector

$$\phi : \mathbb{R} \rightarrow (0, 1)^{n_d}$$
Binary Representation (cont)

- Transform each individual

\[ x = (x_1, \ldots, x_j, \ldots, x_{n_x}) \]

with \( x_j \in \mathbb{R} \) to the binary-valued individual,

\[ b = (b_1, \ldots, b_j, \ldots, b_{n_x}) \]

where

\[ b_j = (b_{(j-1)n_d+1}, \ldots, b_{jn_d}) \]

with \( b_l \in \{0, 1\} \) and the total number of bits, \( n_b = n_x n_d \)
Binary Representation (cont)

- Decoding each \( b_j \) back to a floating-point representation:

\[
\Phi_j(b) = x_{min,j} + \frac{x_{max,j} - x_{min,j}}{2^{nd} - 1} \left( \sum_{l=1}^{nd-1} b_j(n_d-l)2^l \right)
\]

- That is, use the mapping,

\[
\Phi_j : \{0, 1\}^{nd} \rightarrow [x_{min,j}, x_{max,j}]
\]
Binary Representation (cont)

Problems with using a binary representation:

- Conversion from a floating-point value to a bitstring of $n_d$ bits has the maximum attainable accuracy $\frac{x_{max,j} - x_{min,j}}{2^{n_d} - 1}$ for each vector component, $j = 1, \ldots, n_x$
- Binary coding introduces Hamming cliffs:

\[
\begin{align*}
7_{10} &= 0111_2 \text{ vs } 8_{10} = 1000_2 \\
\text{Large change in solution is needed for small change in fitness}
\end{align*}
\]
Chapter 8: Introduction to Evolutionary Computation

Gray Coding

- In Gray coding, the Hamming distance between the representation of successive numerical values is one.
- For 3-bit representations:

<table>
<thead>
<tr>
<th>Binary</th>
<th>Gray</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>011</td>
</tr>
<tr>
<td>011</td>
<td>010</td>
</tr>
<tr>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td>101</td>
<td>111</td>
</tr>
<tr>
<td>110</td>
<td>101</td>
</tr>
<tr>
<td>111</td>
<td>100</td>
</tr>
</tbody>
</table>

Converting binary strings to Gray bit strings:

\[ g_1 = b_1 \]
\[ g_i = b_{i-1} b_i + \overline{b_{i-1}} b_i \]

where \( b_i \) is bit \( i \) of the binary number

\[ b_1 b_2 \cdots b_{n_b} \]
Gray Coding (cont)

A Gray code representation, \( b_j \) can be converted to a floating-point representation using

\[
\Phi_j(b) = x_{min,j} + \frac{x_{max,j} - x_{min,j}}{2^{n_d} - 1} \left( \sum_{l=1}^{n_d-1} \left( \sum_{q=1}^{n_d-l} b_{(j-1)n_d+q} \right) \mod 2 \right) 2^l
\]
Other Representations

Other representations:

- Real-valued representations, where $x_{ij} \in \mathbb{R}$
- Integer-valued representations, where $x_{ij} \in \mathbb{Z}$
- Discrete-valued representations, where $x_{ij} \in \text{dom}(x_j)$
- Tree-based representations as used in Genetic Programming
- Mixed representations
Initialization

- A population of candidate solutions is maintained
- Values are randomly assigned to each gene from the domain of the corresponding variable
- Uniform random initialization is used to ensure that the initial population is a uniform representation of the entire search space
- What are the consequences of the size of the initial population?
  - Computational complexity per generation
  - Number of generations to converge
  - Quality of solutions obtained – Exploration ability
Survival of the Fittest

Based on the Darwinian model, individuals with the best characteristics have the best chance to survive and to reproduce

A *fitness function* is used to determine the ability of an individual of an EA to survive

The fitness function, $f$, maps a chromosome representation into a scalar value:

$$f : \Gamma^{n_x} \rightarrow \mathbb{R}$$

where $\Gamma$ represents the data type of the elements of an $n_x$-dimensional chromosome
Objective Function

- The fitness function represents the objective function, $\Psi$, which represents the optimization problem.
- Sometimes the chromosome representation does not correspond to the representation expected by the objective function, in which case

$$f : S_C \xrightarrow{\Phi} S_X \xrightarrow{\Psi} \mathbb{R} \xrightarrow{\Upsilon} \mathbb{R}_+$$

- $S_C$ represents the search space of the objective function
- $\Phi$ represents the chromosome decoding function
- $\Psi$ represents the objective function
- $\Upsilon$ represents a scaling function
- For example:

$$f : \{0, 1\}^{n_b} \xrightarrow{\Phi} \mathbb{R}^{n_x} \xrightarrow{\Psi} \mathbb{R} \xrightarrow{\Upsilon} \mathbb{R}_+$$
Objective Function Types

Types of objective functions, resulting in different problem types:

- **Unconstrained** objective functions, but still subject to boundary constraints
- **Constrained** objective functions
- **Multi-objective** functions, where more than one objective has to be optimized
- **Dynamic** or **noisy** objective functions
Selection Operators

- Relates directly to the concept of survival of the fittest
- The main objective of selection operators is to emphasize better solutions
- Selection steps in an EA:
  - Selection of the new population
  - Selection of parents during reproduction
- Selective pressure:
  - Takeover time
  - Relates to the time it requires to produce a uniform population
  - Defined as the speed at which the best solution will occupy the entire population by repeated application of the selection operator alone
  - High selective pressure reduces population diversity
Random Selection

- Each individual has a probability of $\frac{1}{n_s}$ to be selected
- $n_s$ is the size of the population
- No fitness information is used
- Random selection has the lowest selective pressure
Proportional Selection

- Biases selection towards the most-fit individuals
- A probability distribution proportional to the fitness is created, and individuals are selected by sampling the distribution,

\[ \varphi_s(x_i(t)) = \frac{f_{\Upsilon}(x_i(t))}{\sum_{l=1}^{n_s} f_{\Upsilon}(x_l(t))} \]

- \( n_s \) is the total number of individuals in the population
- \( \varphi_s(x_i) \) is the probability that \( x_i \) will be selected
- \( f_{\Upsilon}(x_i) \) is the scaled fitness of \( x_i \), to produce a positive floating-point value
Chapter 8: Introduction to Evolutionary Computation

Introduction to Evolutionary Computation

Generic Evolutionary Algorithm

Representation

Initial Population

Fitness Function

Selection

Reproduction Operators

Stopping Conditions

Evolutionary Computation versus Classical Optimization

Proportional Selection: Sampling Methods

- Roulette wheel selection: Assuming maximization and normalized fitness values (Algorithm 8.2):

Let $i = 1$, where $i$ denotes the chromosome index;

Calculate $\varphi_s(x_i)$;

$sum = \varphi_s(x_i)$;

Choose $r \sim U(0, 1)$;

while $sum < r$ do

$i = i + 1$, i.e. advance to the next chromosome;

$sum = sum + \varphi_s(x_i)$;

end

Return $x_i$ as the selected individual;

- High selective pressure
Proportional Selection: Stochastic Universal Sampling

- Uses a single random value to sample all of the solutions by choosing them at evenly spaced intervals
- Like roulette wheel sampling, construct a wheel with sections for each of the $n_s$ chromosomes
- Instead of one hand that is spun once for each sample, a multi-hand is spun just once
- The hand has $n_s$ arms, equally spaced
- The number of times chromosome $i$ is sampled is the number of arms that fall into the chromosomes section of the wheel
Tournament Selection

- Selects a group of $n_{ts}$ individuals randomly from the population, where $n_{ts} < n_s$
- Select the best individual from this tournament
- For crossover with two parents, tournament selection is done twice
- Tournament selection limits the chance of the best individual to dominate
- Large $n_{ts}$ versus small $n_{ts}$
- Selective pressure depends on the value of $n_{ts}$
Rank-Based Selection

- Uses the rank ordering of fitness values to determine the probability of selection
- Selection is independent of absolute fitness
- Non-deterministic linear sampling:
  - Selects an individual, \( x_i \), such that \( i \sim U(0, U(0, n_s - 1)) \)
  - The individuals are sorted in decreasing order of fitness value
  - Rank 0 is the best individual
  - Rank \( n_s - 1 \) is the worst individual
Rank-Based Selection (cont)

- **Linear ranking:**
  - Assumes that the best individual creates $\hat{\lambda}$ offspring, and the worst individual $\tilde{\lambda}$
  - $1 \leq \hat{\lambda} \leq 2$ and $\tilde{\lambda} = 2 - \hat{\lambda}$
  - The selection probability of each individual is calculated as
    \[
    \varphi_s(x_i(t)) = \frac{\tilde{\lambda} + (f_r(x_i(t))/(n_s - 1))(\hat{\lambda} - \tilde{\lambda})}{n_s}
    \]
  - where $f_r(x_i(t))$ is the rank of $x_i(t)$
Rank-Based Selection (cont)

- Nonlinear ranking:

\[ \varphi_s(x_i(t)) = \frac{1 - e^{-f_r(x_i(t))}}{\beta} \]

\[ \varphi_s(x_i) = \nu(1 - \nu)^{np - 1 - f_r(x_i)} \]

- \( f_r(x_i) \) is the rank of \( x_i \)
- \( \beta \) is a normalization constant
- \( \nu \) indicates the probability of selecting the next individual

- Rank-based selection operators may use any sampling method to select individuals
Boltzmann Selection

- Based on the thermodynamical principles of simulated annealing
- Selection probability:

$$\varphi(x_i(t)) = \frac{1}{1 + e^{f(x_i(t))/T(t)}}$$

$T(t)$ is the temperature parameter
- An initial large value ensures that all individuals have an equal probability of being selected
- As $T(t)$ becomes smaller, selection focuses more on the good individuals
- Can use any sampling method
Boltzmann Selection (cont)

- Boltzmann selection can be used to select between two individuals.
- For example, if

\[
U(0, 1) > \frac{1}{1 + e^{(f(x_i(t)) - f(x'_i(t))) / T(t)}}
\]

then \(x'_i(t)\) is selected; otherwise, \(x_i(t)\) is selected.
(\(\mu + \lambda\))-Selection

- Deterministic rank-based selection methods used in evolutionary strategies
- \(\mu\) indicates the number of parents
- \(\lambda\) is the number of offspring produced from each parent
- \((\mu, \lambda)\)-selection selects the best \(\mu\) offspring for the next population
- \((\mu + \lambda)\)-selection selects the best \(\mu\) individuals from both the parents and the offspring
Elitism

- Ensures that the best individuals of the current population survive to the next generation
- The best individuals are copied to the new population without being mutated
- The more individuals that survive to the next generation, the less the diversity of the new population
Hall of Fame

- Similar to the list of best players of an arcade game
- For each generation, the best individual is selected to be inserted into the hall of fame
- Contain an archive of the best individuals found from the first generation
- The hall of fame can be used as a parent pool for the crossover operator
Types of Operators

- **Reproduction**: process of producing offspring from selected parents by applying crossover and/or mutation operators

- **Crossover**: is the process of creating one or more new individuals through the combination of genetic material randomly selected from two or more parents

- **Mutation**:
  - The process of randomly changing the values of genes in a chromosome
  - The main objective is to introduce new genetic material into the population, thereby increasing genetic diversity
  - Mutation probability and step sizes should be small
  - Proportional to the fitness of the individual?
  - Start with large mutation probability, decreased over time?
When to Stop

- Limit the number of generations, or fitness evaluations
- Stop when population has converged:
  - Terminate when no improvement is observed over a number of consecutive generations
  - Terminate when there is no change in the population
  - Terminate when an acceptable solution has been found
  - Terminate when the objective function slope is approximately zero
**EC vs CO**

- **The search process:**
  - CO uses deterministic rules to move from one point in the search space to the next point
  - EC uses probabilistic transition rules
  - EC applies a parallel search of the search space, while CO uses a sequential search

- **Search surface information:**
  - CO uses derivative information, usually first-order or second-order, of the search space to guide the path to the optimum
  - EC uses no derivative information, but fitness information