

**Artificial Intelligence**

Evolutionary Algorithms

# **Lesson 5: Evolutionary Algorithm – Some Theoretical Foundations**

**Vincenzo Piuri**

---

Università degli Studi di Milano

# Contents

- Motivation
- Schema Theorem
- No-Free-Lunch Theorem

# Why Do Evolutionary Algorithms Work?

- Approach
  - Consider schemata (that is, partly specified binary chromosomes)
  - Investigate how the number of chromosomes matching a schema evolve over several generations
  - Objective: rough stochastic statement that describes how a genetic algorithm explores the search space
  - Confinement to
    - Bit strings (chromosomes of zero and one) of a fixed length  $L$
    - Fitness-proportionate selection (Roulette-wheel selection)
    - Bit-mutation (using the mutation probability  $p_m$ )
    - One point-crossover

# Genetic Algorithm

---

## Algorithm 1 Genetic Algorithm

---

**Input:** target function  $F$

```
1:  $t \leftarrow 0$ 
2:  $P(t) \leftarrow$  create population with  $\mu$  individuals          /*  $\mu$  must be even */
3: evaluate  $P(t)$  with  $F$ 
4: while termination criterion is not fulfilled {
5:    $P'(t) \leftarrow$  select  $\mu$  individuals  $A^{(1)}, \dots, A^{(\mu)}$  from  $P(t)$  with roulette wheel selection
6:    $P'' \leftarrow \emptyset$ 
7:   for  $i \leftarrow 1, \dots, \frac{\mu}{2}$  {
8:      $u \leftarrow$  choose random number from  $U([0, 1])$ 
9:     if  $u \leq p_x$  {                                           /* recomb. prob.  $p_x$  */
10:       $B, C \leftarrow$  one-point crossover( $A^{(2i-1)}, A^{(2i)}$ )
11:    } else {
12:       $B \leftarrow A^{(2i-1)}$ 
13:       $C \leftarrow A^{(2i)}$ 
14:    }
15:     $B \leftarrow$  Bit-Mutation( $B$ )
16:     $C \leftarrow$  Bit-Mutation( $C$ )
17:     $P'' \leftarrow P'' \cup \{B, C\}$ 
18:  }
19:  evaluate  $P''$  with  $F$ 
20:   $t \leftarrow t + 1$ 
21:   $P(t) \leftarrow P''$ 
22: }
23: return best individual from  $P(t)$ 
```

---

# Schemata

- Definition: Schema

- A schema  $h$  is a character string of length  $L$  over the alphabet  $\{0, 1, *\}$ , that is  $h \in \{0, 1, *\}^L$
- The character  $*$  is called wildcard character or Don't-Care-Symbol

- Definition: Matching

- A chromosome  $c \in \{0, 1\}^L$  matches a schema  $h \in \{0, 1, *\}^L$ , written as:  $c \triangleleft h$ , if and only if it coincides with  $h$  at all positions where  $h$  is 0 or 1
- Positions at which  $h$  is  $*$  are not taken into account

# Schemata: an Illustration

$h = **0*11*10*$  schema of length 10

$c_1 = 1100111100$  matches schema  $h$ , that is  $c_1 \triangleleft h$

$c_2 = 1111111111$  does not match  $h$ , that is  $c_2 \not\triangleleft h$

# Influence of Selection (1)

- Effect of selection: what fitness the chromosomes have that match a schema  $h$ ?
  - Approach: defining the mean relative fitness of chromosomes
- Definition: Mean Relative Fitness
  - The mean relative fitness of chromosomes that match schema  $h$  in the population  $P(t)$  is

$$f_{\text{rel}}(h) = \frac{\sum_{A \in P(t), A.G \triangleleft h} A.F_{\text{rel}}}{|\{A \in P(t) \mid A.G \triangleleft h\}|}$$

## Influence of Selection (2)

- The average number of chromosomes in the next generation of a Schema  $h$  matching chromosome is

$$f_{\text{rel}}(h) \cdot |P|$$

- The expected number of chromosomes that match schema  $h$  after selection, is

$$(\text{Number of previously matching chromosomes}) \cdot f_{\text{rel}}(h) \cdot |P|$$



## Influence of Selection (3)

- $$\begin{aligned}
 f_{\text{rel}}(h) \cdot |P| &= \frac{\sum_{A \in P(t), A.G \triangleleft h} A.F_{\text{rel}}(c)}{|\{A \in P(t) \mid A.G \triangleleft h\}|} \cdot |P| \\
 &= \frac{\sum_{A \in P(t), A.G \triangleleft h} \frac{A.F}{\sum_{B \in P(t)} B.F}}{|\{A \in P(t) \mid A.G \triangleleft h\}|} \cdot |P| \\
 &= \frac{\frac{\sum_{A \in P(t), A.G \triangleleft h} A.F}{|\{A \in P(t) \mid A.G \triangleleft h\}|}}{\frac{\sum_{B \in P(t)} B.F}{|P|}} = \frac{\overline{f_t(h)}}{\overline{f_t}}
 \end{aligned}$$

- $\overline{f_t(h)}$  average fitness of the chromosomes matching  $h$  in  $P(t)$
- $\overline{f_t}$  average fitness of all chromosomes of the  $t$ -th generation
- The average number of offspring is the ratio of the average fitness of a schema and the total average fitness

# Influence of Mutation (1)

- We need measures with which we can compute probabilities that the match to a schema is preserved
- Definition: Order (for 1-Bit- and Bit-Mutation)
  - The order of a schema  $h$  is the number of zeroes and ones in  $h$ , that is  $\text{ord}(h) = \#0 + \#1 = L - \# *$  ( $\#$ : number of occurrences of)
- For instance:  $\text{ord}(** 0 * 11 * 10 *) = 5$

# Influence of Mutation (2)

- Match to schema  $h$  is preserved
  - With Bit-Mutation using probability  $(1 - p_m)^{\text{ord}(h)}$
  - With 1-Bit-Mutation with
    - Prob.  $1 - \frac{\text{ord}(h)}{L}$ , if bit is inverted,
    - Prob.  $1 - \frac{\text{ord}(h)}{2L}$ , if new bit is determined by random
- Explanation
  - Bit-Mutation inverts a bit with probability  $p_m$  and with probability  $(1 - p_m)$  otherwise
  - 1-Bit-Mutation chooses one of the  $L$  genes of a chromosome of length  $L$  with same probability

## Influence of Crossover (3)

- Definition: Defining Length (for one-point crossover)
  - The defining length of a schema  $h$  is the difference between the position of the last 0/1 and the first 0/1 in  $h$
- Example:  $dl(**0*11*10*) = 9 - 3 = 6$

# Influence of Crossover (4)

- One-point crossover: probability that the cut point splits a chromosome in such a way that some of the fixed characters of a schema lie on one side of the cut and some on the other is  $\frac{dl(h)}{L-1}$
- Explanation:
  - One-point crossover:  $L - 1$  possible cut points on chromosomes of length  $L$  (all equally likely)
  - With  $dl(h)$  of these cut points, genes specified by the schema are exchanged between individuals
  - Matching might (or might not) get lost

# Schema Theorem

# Definitions (1)

- Definition: Expected value of matching chromosomes

$N(h, t)$  is the expected value of the number of chromosomes that match the schema  $h$  during the  $t$ -th generation

- Definition: Expected value after selection

$N(h, t + \Delta t_s)$  is the expected value of the number of chromosomes that match the schema  $h$  during the  $t$ -th generation after selection

## Definitions (2)

- Definition: Expected value after crossover

$N(h, t + \Delta t_s + \Delta t_x)$  is the expected value of the number of chromosomes that match the schema  $h$  during the  $t$ -th generation after selection and crossover

- Definition: Expected value after mutation

$N(h, t + \Delta t_s + \Delta t_x + \Delta t_m) = N(h, t + 1)$  is the expected value of the number of chromosomes that match the schema  $h$  during the  $t$ -th generation after selection, crossover and mutation (and thus during the  $t + 1$ -th generation)

- Search the relation between  $N(h, t)$  and  $N(h, t + 1)$



# Selection

- Impact of selection: can be described by average fitness

$$N(h, t + \Delta ts) = N(h, t) \cdot f_{\text{rel}}(h) \cdot |P|$$

$N(h, t) \cdot f_{\text{rel}}(h)$  probability that chosen chromosome matches schema  $h$

$N(h, t) \cdot f_{\text{rel}}(h)$  average number of offsprings for one chromosome matching schema  $h$

- Note: relative fitness  $f_{\text{rel}}(h)$  cannot be determined exactly, as the number chromosomes matching schema  $h$  is only an approximation

# Crossover (1)

- Impact of Crossover: described by

$$N(h, t + \Delta t_s + \Delta t_x) \\ = \underbrace{(1 - p_x) \cdot N(h, t + \Delta t_s)}_{E_A} + \underbrace{p_x \cdot N(h, t + \Delta t_s) \cdot (1 - p_{\text{loss}})}_{E_B} + C$$

$p_x$  probability of crossover

$p_{\text{loss}}$  probability of a chromosome matching schema  $h$  losing its matching during 1-point-crossover

$E_A$  expected value of the number of chromosomes matching schema  $h$  and not taking part in crossover

$E_B$  expected value of the number of chromosomes taking part in crossover without losing its matching to schema  $h$

$C$  gained number of chromosomes matching schema  $h$

## Crossover (2)

- Substitution of the expression for  $p_{\text{loss}}$  yields

$$\begin{aligned}
 & N(h, t + \Delta t_s + \Delta t_x) \\
 & \geq (1 - p_x) \cdot N(h, t + \Delta t_s) \\
 & \quad + p_x \cdot N(h, t + \Delta t_s) \cdot \left(1 - \frac{dl(h)}{L-1} \cdot \left(1 - \frac{N(h, t + \Delta t_s)}{|P|}\right)\right) \\
 & = N(h, t + \Delta t_s) \left(1 - p_x + p_x \cdot \left(1 - \frac{dl(h)}{L-1} \cdot \left(1 - \frac{N(h, t + \Delta t_s)}{|P|}\right)\right)\right) \\
 & = N(h, t + \Delta t_s) \cdot \left(1 - p_x \frac{dl(h)}{L-1} \cdot \left(1 - \frac{N(h, t + \Delta t_s)}{|P|}\right)\right) \\
 & \stackrel{(*)}{=} N(h, t) \cdot f_{\text{rel}}(h) \cdot |P| \cdot \left(1 - p_x \frac{dl(h)}{L-1} \cdot (1 - N(h, t) \cdot f_{\text{rel}}(h))\right)
 \end{aligned}$$

- Step (\*): twice use of the previously derived relation

$$N(h, t + \Delta t_s) = N(h, t) \cdot f_{\text{rel}}(h) \cdot |P|$$

# Mutation

- Impact of binary mutation: described by order

$$\begin{aligned} N(h, t + 1) &= N(h, t + \Delta t_s + \Delta t_x + \Delta t_m) \\ &= N(h, t + \Delta t_s + \Delta t_x) \cdot (1 - p_m)^{\text{ord}(h)} \end{aligned}$$

- Explanation: in order to not lose matching, none of the  $\text{ord}(h)$  genes fixed in schema  $h$  must be altered
- Alternative models possible, e.g.: exactly one bit altered per chromosome  $\Rightarrow$  1 bit mutation

$$\begin{aligned} N(h, t + 1) &= N(h, t + \Delta t_s + \Delta t_x + \Delta t_m) \\ &= N(h, t + \Delta t_s + \Delta t_x) \cdot \left(1 - \frac{\text{ord}(h)}{L}\right) \end{aligned}$$

# The Schema Theorem

- Including binary mutation, the following holds

$$N(h, t + 1) \geq f_{\text{rel}}(h) \cdot |P| \cdot \left( 1 - p_x \frac{\text{dl}(h)}{L - 1} \cdot (1 - N(h, t) \cdot f_{\text{rel}}(h)) \right) \cdot (1 - p_m)^{\text{ord}(h)} \cdot N(h, t)$$

- Substitution of the fitness relations yields

$$N(h, t + 1) \geq \frac{\overline{f_t(h)}}{\overline{f_t}} \left( 1 - p_x \frac{\text{dl}(h)}{L - 1} \left( 1 - \frac{N(h, t)}{|P|} \cdot \frac{\overline{f_t(h)}}{\overline{f_t}} \right) \right) \cdot (1 - p_m)^{\text{ord}(h)} \cdot N(h, t)$$

*Interpretation: schemata with*

- *a score above average*
- *short defining length*
- *low order*

*breed very heavily (approximately exponential)*

# Building Block Hypothesis

- A Genetic Algorithms explores  $\Omega$  particularly well for schemata with good average fitness, low defining length, and low order
- Chromosomes breed particularly well in these regions
- These schemata are called building blocks, thus the name Building Block Hypothesis
- Note:
  - This form of such hypothesis only holds for bit sequences, fitness proportional selection, binary mutation and 1-point crossover
  - With other genetic operations the blocks can possibly be described by other characteristics
  - High average fitness is always a good characteristic, as every selection method prefers chromosomes with high fitness

# No Free Lunch Theorem: Preconditions

- Search space  $\Omega$
- $\mathcal{F}$  space of all optimization problems (objective function)
- Lack of knowledge regarding optimization problem
  - Uniform distribution among all these problems
  - Every problem  $F \in \mathcal{F}$  occurs with a probability of  $\frac{1}{|\mathcal{F}|}$
- Further simplifications
  - $\forall F \in \mathcal{F} \quad F: \Omega \mapsto \mathbb{R}$  holds
  - $\forall F \in \mathcal{F}$  are defined on the same search space  $\Omega$
- Let  $A$  be the set of all optimization algorithms operating on  $\Omega$

# No Free Lunch Theorem: Characterization of an Algorithm (1)

- Which individuals are considered in which order on  $F \in \mathcal{F}$
- Only  $n$  evaluations possible for optimization
  - $\text{Optimization}_{F,n}: \mathcal{A} \mapsto \Omega^n$
- With every optimization the algorithm assesses an individual  
 $\Rightarrow \text{Optimization}_{F,n}(\text{Alg})$  contains  $n$  different individuals
- Let every algorithm Alg be deterministic  
 $\Rightarrow \text{Optimization}_{F,n}(\text{Alg})$  unique
- For a problem  $F \in \mathcal{F}$ , an optimization problem  $\text{Alg} \in \mathcal{A}$  and  $n \in \mathbb{N}$

$$\text{Optimization}_{F,n}(\text{Alg}) = (y_1, \dots, y_n) \in \Omega^n$$

with  $y_i \neq y_j$  for  $i \neq j$  and  $y_k$  being the individual, which analyzes Alg with  $F$  as  $k$ -th element



# No Free Lunch Theorem: Characterization of an Algorithm (2)

- Comparison of  $\text{Alg}_1, \text{Alg}_2 \in \mathcal{A}$ 
  - Performance schemata  $\text{QuAlg}$  (quality of an algorithm) defined with  $\text{Optimization}_{F,n}(\text{Alg}) = (y_1, \dots, y_n)$  by  $q_n: \mathbb{R}^n \mapsto \mathbb{R}$  as  $\text{QuAlg}_{F,n}(\text{Alg}) = (F(y_1), \dots, F(y_n))$ 
    - average / best goodness
    - number of needed evaluations until optimum is reached
  - Expected performance  $E$  of the first  $n$  evaluations of  $\text{Alg}$  on an arbitrary problem

$$E \left[ \text{QuAlg}_{F,n}(\text{Alg}) \mid F \in \mathcal{F} \right] = \frac{1}{\#\mathcal{F}} \sum_{F \in \mathcal{F}} \text{QuAlg}_{F,n}(\text{Alg})$$

$\Rightarrow$  average on all possible problems

# No Free Lunch Theorem

- Theorem: No Free Lunch

For two algorithms  $Alg_1, Alg_2 \in \mathcal{A}$  and the class of all problems  $\mathcal{F}$  the following applies to performance schema  $QuAlg$

$$E \left[ QuAlg_{F,n}(Alg_1) \mid F \in \mathcal{F} \right] = E \left[ QuAlg_{F,n}(Alg_2) \mid F \in \mathcal{F} \right]$$

# No Free Lunch Theorem: Consequences (1)

- No algorithm is superior to all others on average

- If there is an algorithm superior on  $\mathcal{F}' \subset \mathcal{F}$ , thus

$$E \left[ \text{QuAlg}_{F,n}(\text{Alg}_1) \mid F \in \mathcal{F}' \right] < E \left[ \text{QuAlg}_{F,n}(\text{Alg}_2) \mid F \in \mathcal{F}' \right]$$

then it follows that

$$E \left[ \text{QuAlg}_{F,n}(\text{Alg}_1) \mid F \in \mathcal{F} \setminus \mathcal{F}' \right] > E \left[ \text{QuAlg}_{F,n}(\text{Alg}_2) \mid F \in \mathcal{F} \setminus \mathcal{F}' \right]$$

- For every algorithm,  $\exists$  niche within the space of all problems which it is particularly good for
- Which algorithm to use for which problem?
- Which class of problems is best for a particular algorithm?

# No Free Lunch Theorem: Consequences (2)

- If there is no previous knowledge regarding the problem
  - ⇒ Expected results of an Evolutionary Algorithm are not better than those of any other method
- If there is knowledge regarding the problem
  - e.g. assumptions about the goodness space
  - ⇒ General application of certain algorithms is suggested
- Knowledge regarding structure of the problem should influence the choice or design of an optimization algorithm