

Stochastic optimization algorithms

Lecture 5, 20200909

Evolutionary algorithms:
Components of EAs

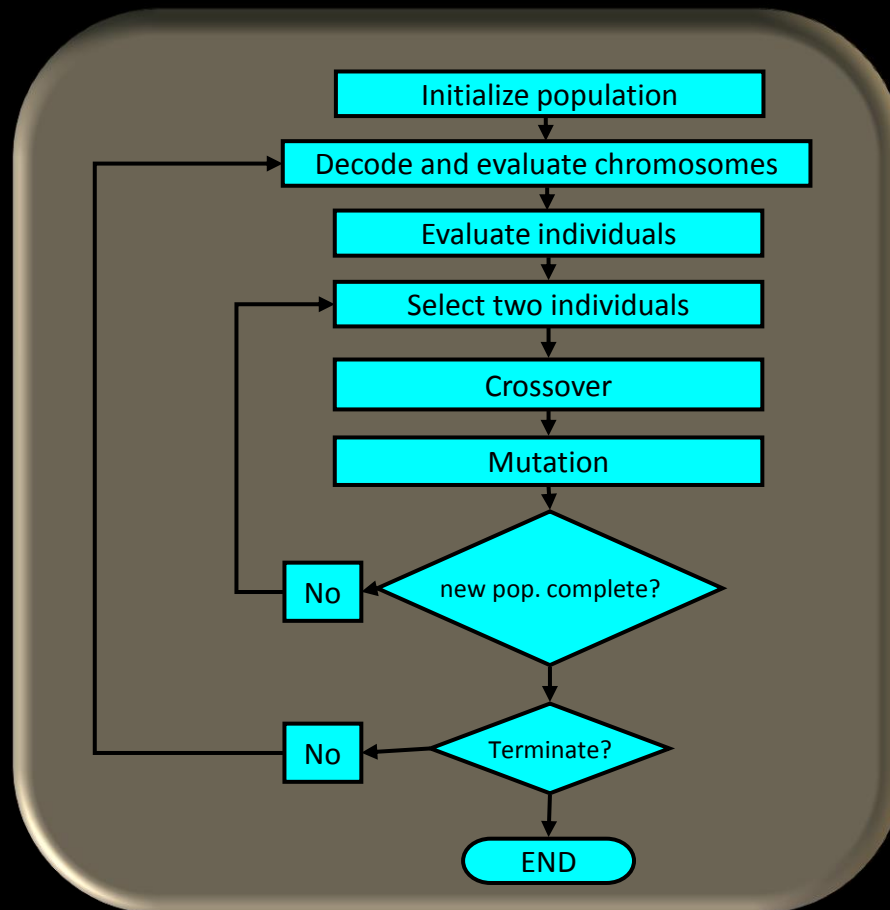
Information

- You do not need to hand in any files from yesterday's Matlab exercise.
- You must, however, hand in solutions to the following:
 - Introductory programming problem (deadline: 20200911, this Friday!)
 - Home problem 1 (deadline: 20200922)
- All relevant documents (including instructions – read carefully!) are available on the course web page.
- Note: Check the FAQ frequently. It is located in the module called *Miscellaneous*.

Today's learning goals

- After this lecture you should be able to
 - Describe and compare some different encoding schemes
 - Describe and compare two different selection methods
 - Describe the crossover operator
 - Describe the process of mutation
 - Describe the processes of replacement and elitism
 - Define a standard genetic algorithm

From last time...



pp. 46-48

Encoding schemes

- Binary encoding

1	0	0	1	0	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---

$$x = -d + \frac{2d}{1 - 2^{-k}}(2^{-1}g_1 + \dots + 2^{-k}g_k),$$

Encoding schemes

- Binary encoding

1	0	0	1	0	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---

$$x = -d + \frac{2d}{1 - 2^{-k}} \underbrace{(2^{-1}g_1 + \dots + 2^{-k}g_k)},$$

Encoding schemes

- Binary encoding

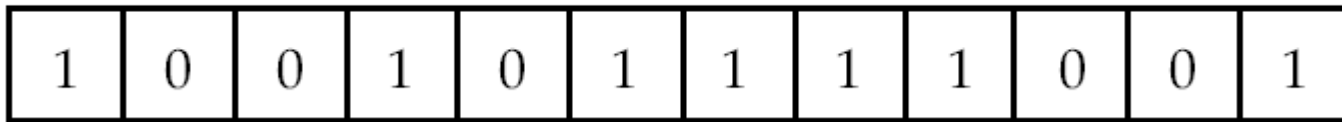
1	0	0	1	0	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---

$$x = -d + \frac{2d}{1 - 2^{-k}} \underbrace{(2^{-1}g_1 + \dots + 2^{-k}g_k)}_{\sum_{j=1}^k 2^{-j}}$$

Maximum value (if all alleles are equal to 1): $\sum_{j=1}^k 2^{-j} = 1 - 2^{-k}$

Encoding schemes

- Binary encoding

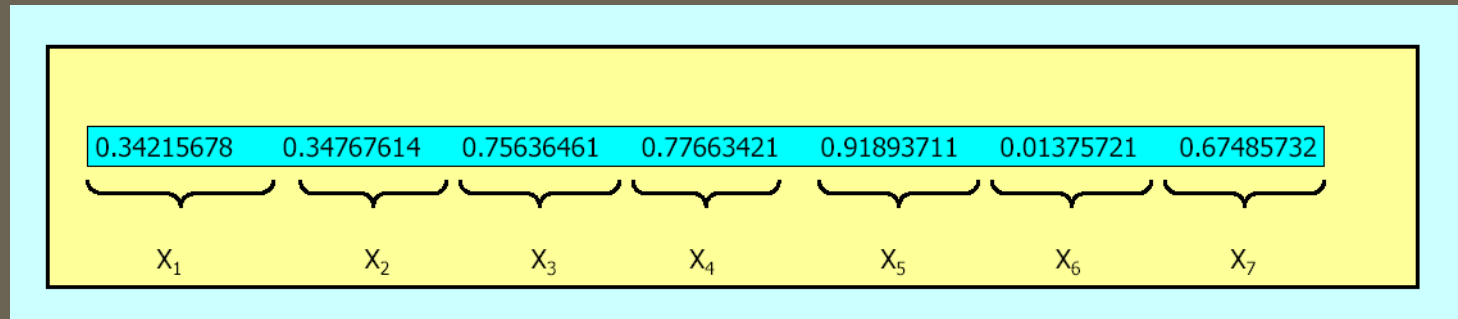


$$x = -d + \frac{2d}{1 - 2^{-k}} \underbrace{(2^{-1}g_1 + \dots + 2^{-k}g_k)}_{\sum_{j=1}^k 2^{-j} = 1 - 2^{-k}}, \quad \in [-d, d]$$

Maximum value (if all alleles are equal to 1): $\sum_{j=1}^k 2^{-j} = 1 - 2^{-k}$

Encoding schemes

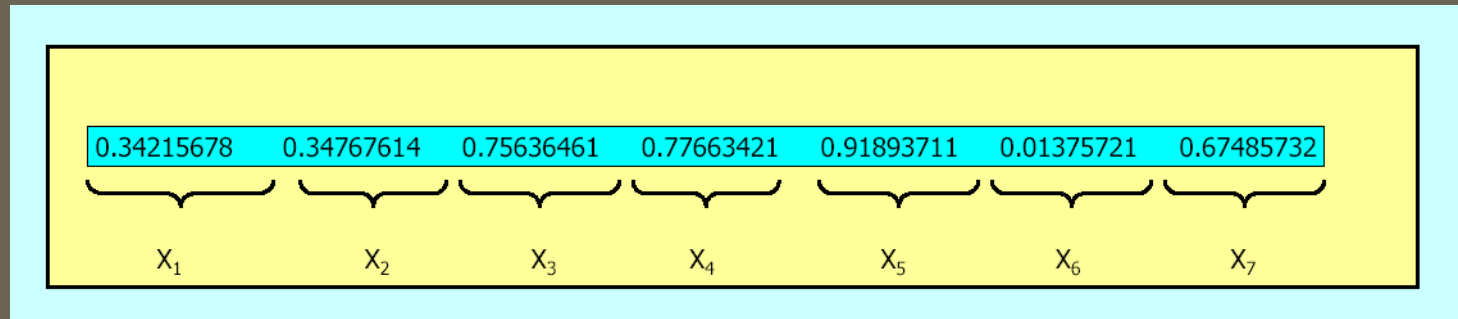
- Real-number encoding



$$x = -d + 2dg$$

Encoding schemes

- Real-number encoding



$$x = -d + 2dg \in [-d, d]$$

Encoding schemes

- Other encoding schemes exist ...
 - Consider a case (with a single variable, for simplicity) where the best chromosome = 1000...00 (=0.5 before rescaling), but the population has converged to 0111...1 (=0.499... before rescaling).
- Gray coding:
 - Small change in genotype => small change in phenotype
 - Standard binary coding : (00) \Leftrightarrow 0, (01) \Leftrightarrow 1, (10) \Leftrightarrow 2, (11) \Leftrightarrow 3
 - Gray coding: (00) \Leftrightarrow 0, (01) \Leftrightarrow 1, (11) \Leftrightarrow 2, (10) \Leftrightarrow 3

Encoding schemes

- Other encoding schemes exist ...
 - Consider a case (with a single variable, for simplicity) where the best chromosome = 1000...00 (=0.5 before rescaling), but the population has converged to 0111...1 (=0.499... before rescaling).
- Gray coding:
 - Small change in genotype => small change in phenotype
 - Standard binary coding : (00) \Leftrightarrow 0, (01) \Leftrightarrow 1, (10) \Leftrightarrow 2, (11) \Leftrightarrow 3
 - Gray coding: (00) \Leftrightarrow 0, (01) \Leftrightarrow 1, (11) \Leftrightarrow 2, (10) \Leftrightarrow 3

Encoding schemes

- Other encoding schemes exist ...
 - Consider a case (with a single variable, for simplicity) where the best chromosome = 1000...00 (=0.5 before rescaling), but the population has converged to 0111...1 (=0.499... before rescaling).
- Gray coding:
 - Small change in genotype => small change in phenotype
 - Standard binary coding : (00) \Leftrightarrow 0, (01) \Leftrightarrow 1, (10) \Leftrightarrow 2, (11) \Leftrightarrow 3
 - Gray coding: (00) \Leftrightarrow 0, (01) \Leftrightarrow 1, (11) \Leftrightarrow 2, (10) \Leftrightarrow 3

Encoding schemes

- Other encoding schemes exist ...
 - Consider a case (with a single variable, for simplicity) where the best chromosome = 1000...00 (=0.5 before rescaling), but the population has converged to 0111...1 (=0.499... before rescaling).
- Gray coding:
 - Small change in genotype => small change in phenotype
 - Standard binary coding : (00) \Leftrightarrow 0, (01) \Leftrightarrow 1, (10) \Leftrightarrow 2, (11) \Leftrightarrow 3
 - Gray coding: (00) \Leftrightarrow 0, (01) \Leftrightarrow 1, (11) \Leftrightarrow 2, (10) \Leftrightarrow 3

Encoding schemes

- Other encoding schemes exist ...
 - Messy encoding: Less position-dependent
 - Encodes both position and allele

$$c = ((p_1, g_{p_1}), (p_2, g_{p_2}), \dots, (p_j, g_{p_j}), \dots, (p_m, g_{p_m})),$$

- Problem: avoiding duplicates and missing genes
 - Duplicates: Use the first copy
 - Missing genes: Extend chromosome

Messy encoding (example)

- Messy chromosome $c = ((1,0),(5,1),(3,0),(4,1),(2,1)) \Rightarrow$
standard chromosome: _ _ _ _ _

Messy encoding (example)

- Messy chromosome $c = ((1,0),(5,1),(3,0),(4,1),(2,1)) \Rightarrow$
standard chromosome: 0 — — — —

Messy encoding (example)

- Messy chromosome $c = ((1,0),(5,1),(3,0),(4,1),(2,1)) \Rightarrow$
standard chromosome: 0 — — — 1

Messy encoding (example)

- Messy chromosome $c = ((1,0),(5,1),(3,0),(4,1),(2,1)) \Rightarrow$
standard chromosome: 0 1 0 1 1

Encoding schemes

- Other encoding schemes exist ...
 - Messy encoding: Less position-dependent
 - Encodes both position and allele

$$c = ((p_1, g_{p_1}), (p_2, g_{p_2}), \dots, (p_j, g_{p_j}), \dots, (p_m, g_{p_m})),$$

-
- Problem: avoiding duplicates and missing genes
 - Duplicates: Use the first copy
 - Missing genes: Extend chromosome

Encoding schemes

- Other encoding schemes exist ...
 - Messy encoding: Less position-dependent
 - Encodes both position and allele

$$c = ((p_1, g_{p_1}), (p_2, g_{p_2}), \dots, (p_j, g_{p_j}), \dots, (p_m, g_{p_m})),$$

- Problem: avoiding duplicates and missing genes
 - Duplicates: Use the first copy
 - Missing genes: Extend chromosome

Today's learning goals

- After this lecture you should be able to
 - Describe and compare some different encoding schemes
 - Describe and compare two different selection methods
 - Describe the crossover operator
 - Describe the process of mutation
 - Describe the processes of replacement and elitism
 - Define a standard genetic algorithm



Selection

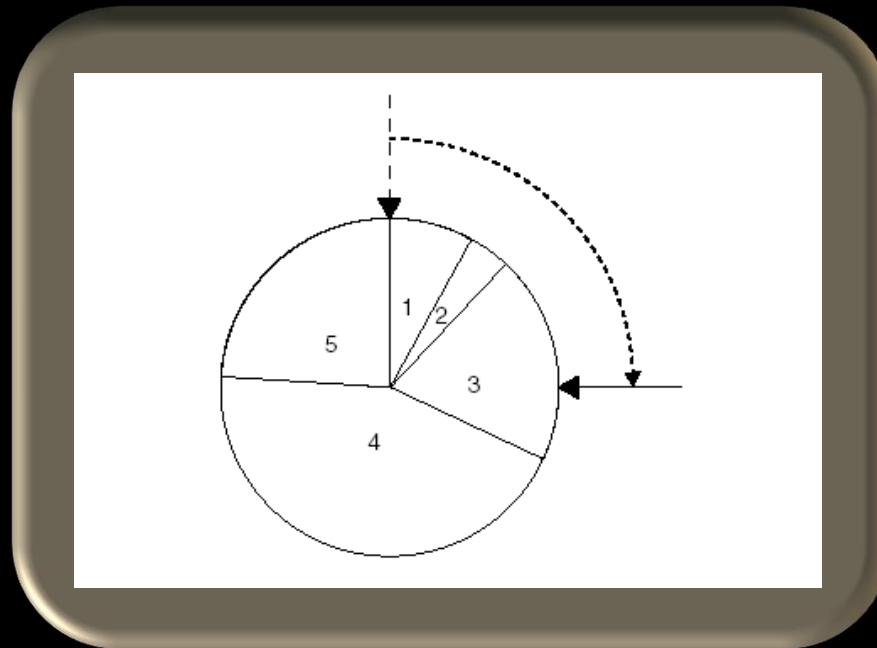
- Two main methods
 - Roulette-wheel selection
 - Tournament selection

Roulette-wheel selection

- Equivalent to spinning a roulette-wheel, on which the size of the sector occupied by each individual is proportional to its fitness:

Roulette-wheel selection

- Equivalent to spinning a roulette-wheel, on which the size of the sector occupied by each individual is proportional to its fitness:



pp. 48-52

Roulette-wheel selection

- In equation form: Select the individual with the smallest j that satisfies

$$\phi_j \equiv \frac{\sum_{i=1}^j F_i}{\sum_{i=1}^N F_i} > r,$$

...where r is a random number in $[0,1[$.

Roulette-wheel selection

- In equation form: Select the individual with the smallest j that satisfies

$$\phi_j \equiv \frac{\sum_{i=1}^j F_i}{\sum_{i=1}^N F_i} > r,$$

...where r is a random number in $[0,1[$.

Roulette-wheel selection

- In addition to the roulette-wheel metaphor, one can also visualize the process on a line from [0,1]:

$$\phi_1 = \frac{F_1}{\sum_{i=1}^N F_i}$$



Roulette-wheel selection

- In addition to the roulette-wheel metaphor, one can also visualize the process on a line from [0,1]:

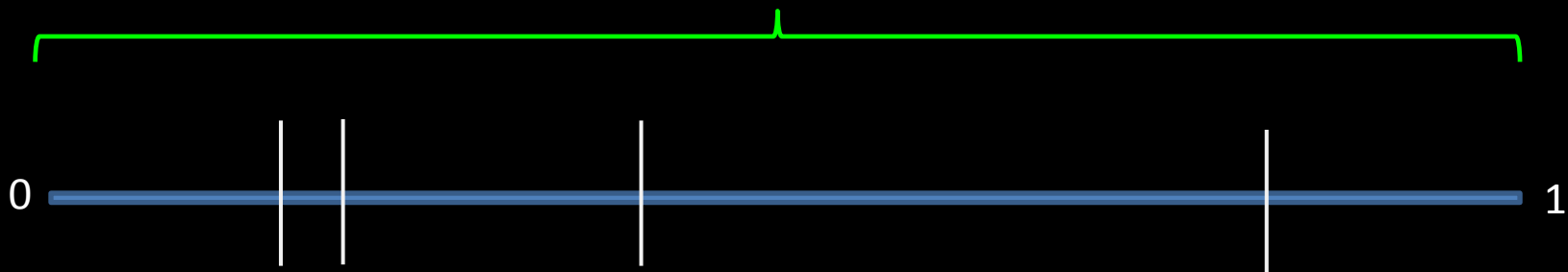
$$\phi_2 = \frac{F_1 + F_2}{\sum_{i=1}^N F_i}$$



Roulette-wheel selection

- In addition to the roulette-wheel metaphor, one can also visualize the process on a line from [0,1]:

$$\phi_N = \frac{\sum_{i=1}^N F_i}{\sum_{i=1}^N F_i} = 1$$



Roulette-wheel selection

- Example 3.3 (p. 49)
- Fitness sum = 25.0 in this example.
 - Individual 1: $F_1 = 2.0 \Rightarrow \phi_1 = \frac{2}{25} = 0.08$
 - Individual 2: $F_2 = 1.0 \Rightarrow \phi_2 = \frac{1+2}{25} = 0.12$ etc.
 - Individual 3: $F_3 = 5.0 \Rightarrow \phi_3 = \frac{1+2+5}{25} = 0.32$ etc.
 - Individual 4: $F_4 = 11.0$
 - Individual 5: $F_5 = 6.0$
- If $r = 0.25$ (say), then $j = 3$ (smallest j such that $\phi_j > r$)

pp. 48-52

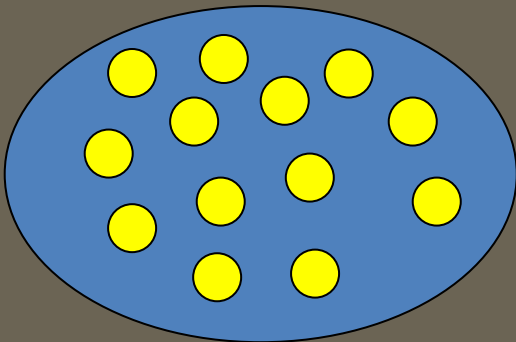


Tournament selection

- Procedure for selection one individual:
 1. Pick two individuals randomly from the population (equal probability for all individuals in the population)
 2. With probability p_{tour} (usually around 0.7-0.8) pick the better of the two individuals (higher fitness). Otherwise, pick the worse of the two.

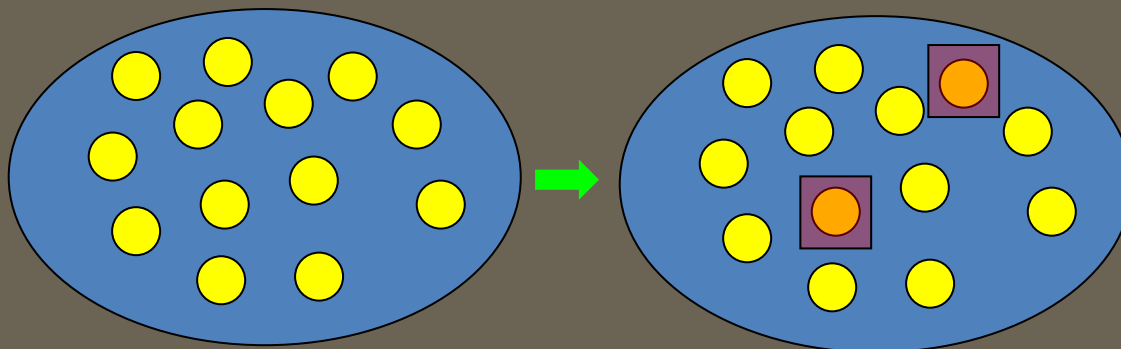
Tournament selection

- Procedure for selection one individual:
 1. Pick two individuals randomly from the population (equal probability for all individuals in the population)
 2. With probability p_{tour} (usually around 0.7-0.8) pick the better of the two individuals (higher fitness). Otherwise, pick the worse of the two.



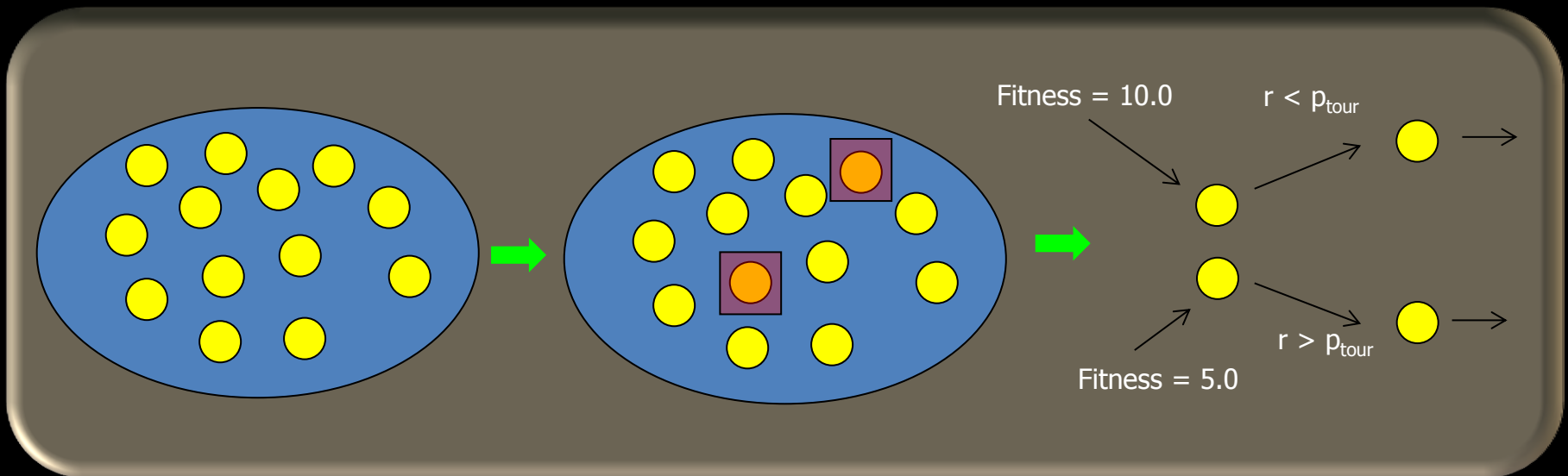
Tournament selection

- Procedure for selection one individual:
 1. Pick two individuals randomly from the population (equal probability for all individuals in the population)
 2. With probability p_{tour} (usually around 0.7-0.8) pick the better of the two individuals (higher fitness). Otherwise, pick the worse of the two.



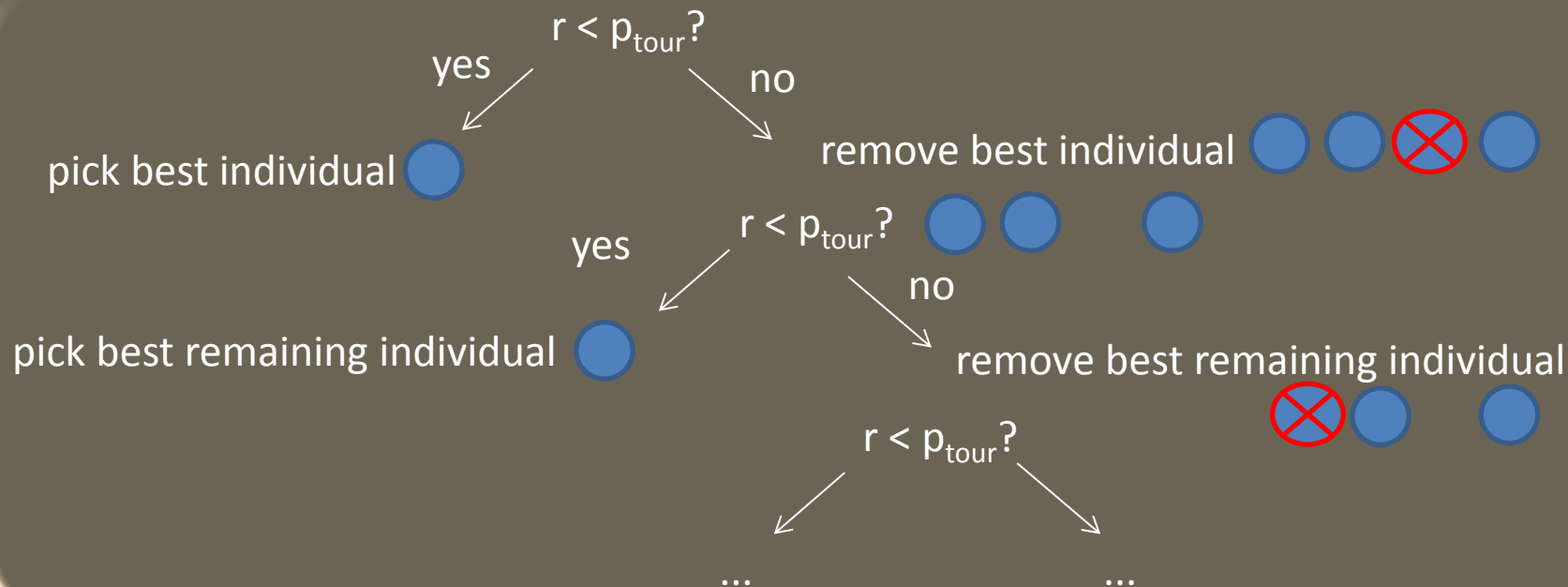
Tournament selection

- Procedure for selection one individual:
 1. Pick two individuals randomly from the population (equal probability for all individuals in the population)
 2. With probability p_{tour} (usually around 0.7-0.8) pick the better of the two individuals (higher fitness). Otherwise, pick the worse of the two.



Tournament selection

- Can be generalized to larger tournaments (see p. 50).



pp. 48-52

Tournament selection

- Note that the procedures described on the previous slides result in *one* individual being selected.
- Thus, selection must be carried out twice to select a pair of individuals.
- Note also that selected individuals are returned to the population after selection. Thus, the same individual can be selected many times, in both roulette-wheel and tournament selection.

Premature convergence

- A situation where the population converges to a suboptimal solution.
- Common problem in evolutionary algorithms.
- For roulette-wheel selection, one can apply so called fitness ranking to reduce the probability of premature convergence.
- We will look more at this topic in a later lecture.

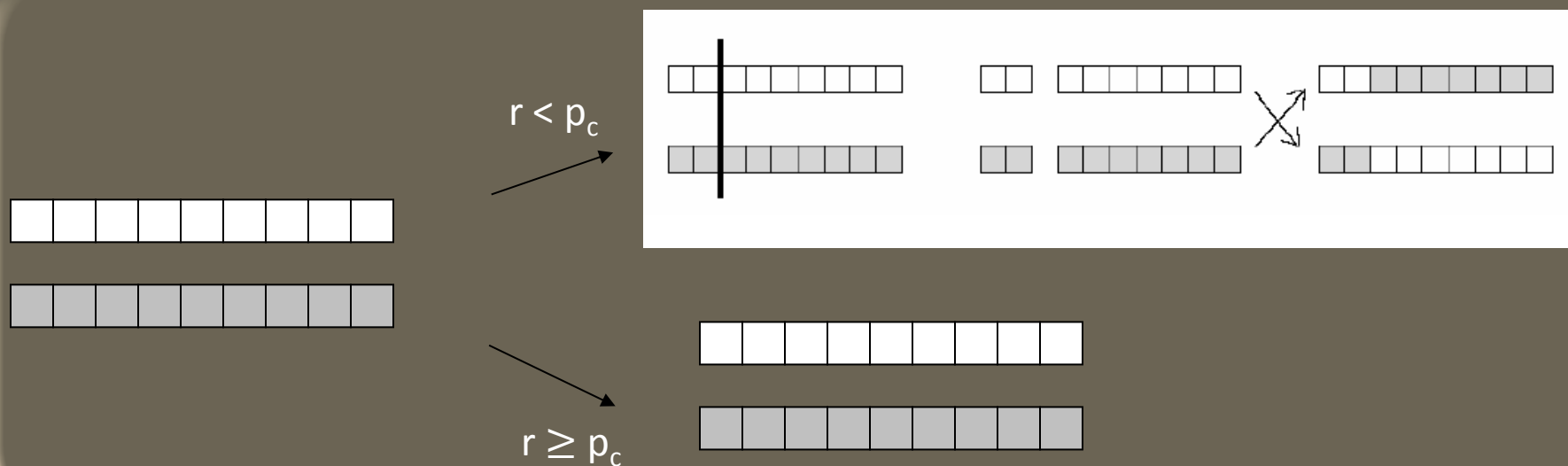
Today's learning goals

- After this lecture you should be able to
 - Describe and compare some different encoding schemes
 - Describe and compare two different selection methods
 - Describe the crossover operator
 - Describe the process of mutation
 - Describe the processes of replacement and elitism
 - Define a standard genetic algorithm



Crossover

- Carried out with a probability p_c . Otherwise: No change in the selected individuals (in this step).



Today's learning goals

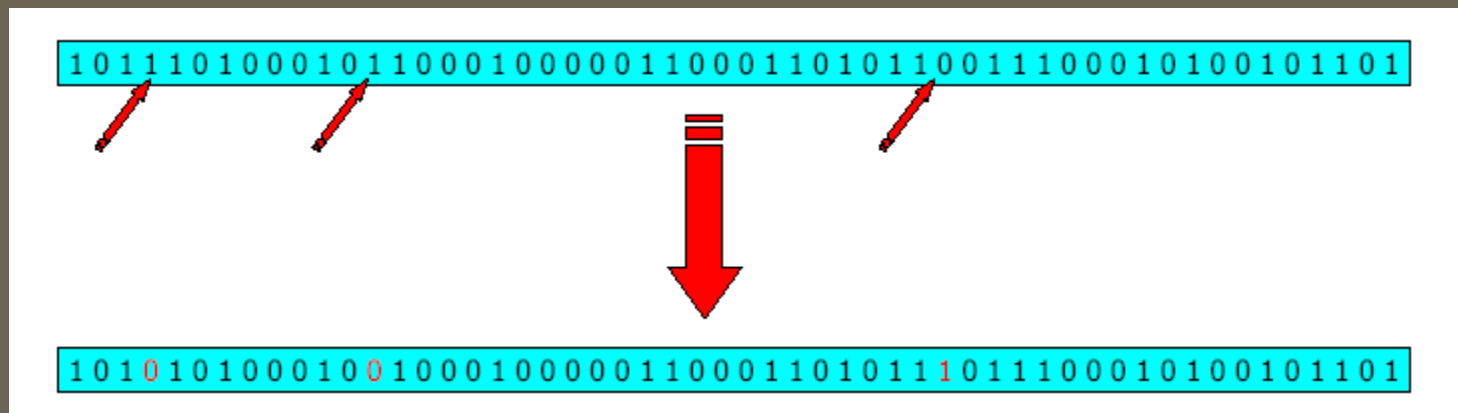
- After this lecture you should be able to
 - Describe and compare some different encoding schemes
 - Describe and compare two different selection methods
 - Describe the crossover operator
 - Describe the process of mutation
 - Describe the processes of replacement and elitism
 - Define a standard genetic algorithm



Mutation

- Small random changes, provide new material for the evolutionary process.
- Mutation rate (per gene): p_{mut} .
- Typical value: $p_{\text{mut}} = \frac{1}{m}$, (m = number of genes).

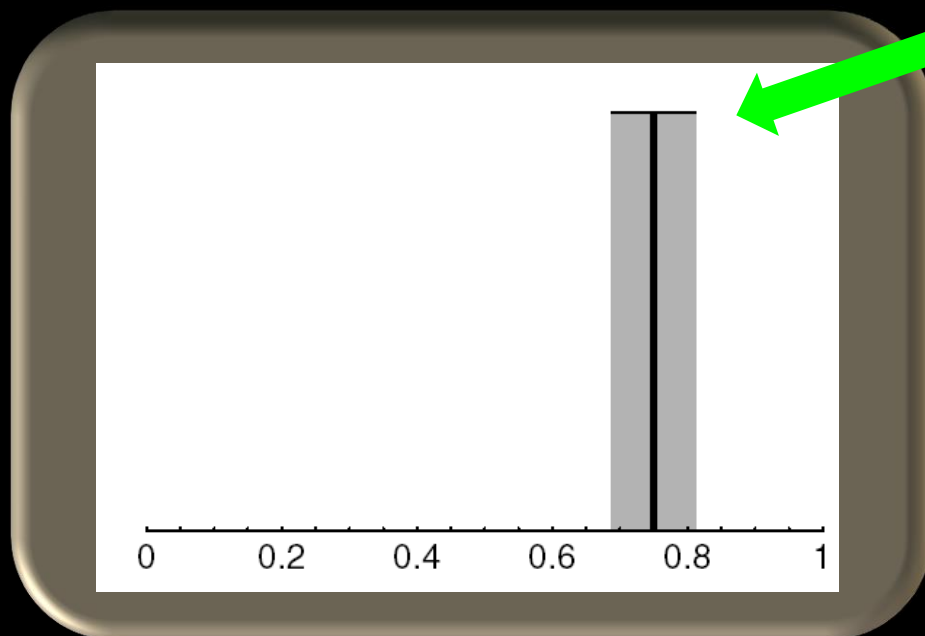
Mutation



- Every gene is checked. If, for a given gene, $r < p_{\text{mut}}$, the gene mutates ($r \in [0,1[$).

Creep mutations

- Used in connection with real-number encoding.
- The new values are drawn randomly from a distribution centered on the old value:



Uniform
creep mutation

pp. 53-55

Creep mutations: Procedure

- For each gene, check (as usual) whether or not the gene should be mutated (using the mutation rate)
 - If the gene should be mutated, check whether or not creep mutations should be used (using p_{creep})
 - If yes, carry out a creep mutation:
$$g \Rightarrow g' = g - C_r/2 + C_r r$$
 - If no, carry out an ordinary mutation:
$$g \Rightarrow g' = r$$
 - ...where, in both cases, r is another random number in $[0,1[$.
 - If the gene should not be mutated, do nothing (proceed to the next gene).

pp. 53-55



Today's learning goals

- After this lecture you should be able to
 - Describe and compare some different encoding schemes
 - Describe and compare two different selection methods
 - Describe the crossover operator
 - Describe the process of mutation
 - Describe the processes of replacement and elitism
 - Define a standard genetic algorithm



Replacement

- **Generational replacement:**
 N new individuals are formed, and they replace the N old individuals (used e.g. in HP1.3).
- **Steady-state replacement:**
New individuals are formed in pairs, and replace the two worst individuals in the population.

Elitism

- A few exact copies (usually one) of the best individual in generation $g - 1$ are copied unchanged to generation g .
- This procedure ensures that
 - ... the maximum fitness will rise monotonously (or at least never fall),
 - ... the best individual is always available when the algorithm is terminated.

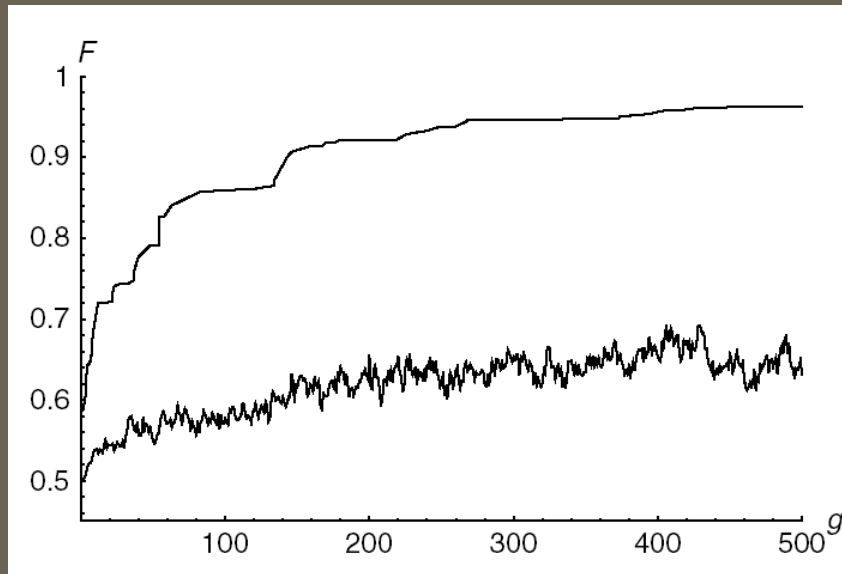
Today's learning goals

- After this lecture you should be able to
 - Describe and compare some different encoding schemes
 - Describe and compare two different selection methods
 - Describe the crossover operator
 - Describe the process of mutation
 - Describe the processes of replacement and elitism
 - Define a standard genetic algorithm



Typical fitness curve

- Top curve: Maximum fitness, Bottom curve: average fitness



Standard GA

- Algorithm 3.2 in the book.
- Study carefully! (Note, however, that elitism can be done more easily by simply copying the best individual into position 1 (in the population) at the end of each generation; see also the Matlab introduction).
- The selection step is usually carried out with tournament selection.

Today's learning goals

- After this lecture you should be able to
 - Describe and compare some different encoding schemes
 - Describe and compare two different selection methods
 - Describe the crossover operator
 - Describe the process of mutation
 - Describe the processes of replacement and elitism
 - Define a standard genetic algorithm



NOTE!

- When solving HP1.3, you can, *to some extent*, make use of the code from the Matlab introduction, but you need to adapt (generalize) the code a bit. Read the instructions carefully!
- We aim to return the results from the introductory programming problem (IPP) as soon as possible. The results will be given in Canvas.
- Make sure to study your results for the IPP before handing in HP1!