

**Autonomous Intelligent Systems,  
Institute for Computer Science VI, University of Bonn**

Dr. N. Goerke

Friedrich-Ebert-Allee 144, 53113 Bonn, Tel: (0228) 73-4167

E-Mail: goerke@ais.uni-bonn.de

www.ais.uni-bonn.de

**Exercises for Artificial Life (MA-INF 4201), SS15**

**Exercises sheet 7, till: Mon 8.6.2015**

1.6.2015

Group	Name	44	45	46	47	48	49	$\Sigma$

**Assignment 44** (3 Points)

How many different offspring can be created from the two parents **A** and **B** by using the 2-point-cross-over recombination-operator?

Both genomes (**A** and **B**) are  $L$ -dimensional vectors that consist of  $L = 64$  independent real values; the individual values are not changed by the 2-point-cross-over operator.

**Assignment 45** (2 Points)

Explain how a hypercube and a binary genome of an Evolutionary Algorithm are related to each other. Draw a sketch, visualizing this for a binary genome that has more than two bits.

**Assignment 46** (2 Points)

An evolutionary algorithm is proposed to find a solution for a salesman who has to plan a cyclic journey, visiting all cities from a given set of  $L$  cities exactly once. For this application the genome has been structured as a sequence of all  $L$  cities to visit.

An easy to implement mutation operator would be to swap two cities within this list.

Is it possible to generate all possible sequences of cities by this mutation operator?

**Assignment 47** (4 Points)

Within an Evolutionary Algorithm a parent individual  $X(i)$  with a genome of  $L$  bit has created  $N$  offspring  $X(i) = Y(i)_n$  identical to the parent  $X(i)$ .

To yield the new generation  $Y(i+1)_n$  the mutation operator is modifying each of these  $N$  offspring  $Y(i)_n$  by flipping each of the  $N * L$  bits with a probability of  $p$ .

Derive a formula that calculates the propability  $Q$  for the case that **none** of the  $N$  new individuals  $Y(i+1)_n$  is identical to the parent  $X(i)$ .

Use your formula, and calculate the probability  $Q$  for the case that  $N = 20$ ,  $L = 100$  and  $p = 0.01$ .

## Assignment 48 (1 Point)

Explain the difference between the two EA steps *external selection* and *parent selection*.

## Assignment 49 (3 Points)

Imagine a high dimensional (e.g.  $N$ -dimensional) unit-hypercube (edge length 1.0 in each dimension).

Imagine further, a set of random vectors  $\mathbf{X}_i$  ( $i = 1, \dots, Z$ ), where each of the components  $x(n, i)$  is a random value between 0.0 and 1.0, drawn independent of each other.

How long is the average distance between two of these random vectors  $\mathbf{X}_i$  and  $\mathbf{X}_j$ ?

How is the distribution of the distances?

What about the distribution of angles between these vectors?

The proposed way to answer the questions, is to implement a small program, with  $N = 1000$  and  $Z = 2000$ .

It has been told to be extremely difficult, to calculate this analytically (not recommended!).

## Programming Assignment PA-D (5 Points, due date 8.6.15)

Write a C, C++, Java or Python Programm, that implements a mini-version of an Evolutionary Algorithm. Please implement or foresee a slot for every essential step of the EA, even if you do not implement it right now.

The Evolutionary Algorithm shall find the maximum of a function  $f(g)$ . Implement a mini population of  $P = 2$  individuals, each with a genome  $g$  that is a vector of real valued  $L$  components.

The external selection steps keeps only the best one of those two individuals, and discards the other one. The implementation of parent selection can thus be omitted, but should be foreseen within the program code. The inheritance step is just copying of the surviving genome to be the newly created genome.

The mutation step is only performed for the newly created genome. Implement the mutation step by just adding a small random vector  $\mathbf{r}$  to the genome, each of the  $L$  components of  $\mathbf{r}$  shall be either equally distributed between  $-\varepsilon$  and  $+\varepsilon$  or normally distributed with adjustable bounds.

Fitness evaluation is just calculating the fitness function  $f(g)$  for all  $P$  genomes of the Population, and providing the results for the external selection process.

Initialize both genomes with random values, and implement a sensefull stopping criterion.

Print out the fitness values of both individuals while the EA is running, and in the end, print the genome of the resulting individual.

Make sure your program is running correctly, and that you have sensible and resonable comments in your sourcecode.