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 11, till: Mon 6.7.2015

29.6.2015

Group	Name	70	71	72	73	74	75	76	$\sum$

Assignment 70 (3 Points)

What are the major differences between the following groups of animals: *flocks*, *herds* and *swarms*?
What difference with respect to the behavior of the individuals can you determine in this groups?

Assignment 71 (2 Points)

In the Didabot experiment the Didabots are creating so called *heaps* or *Clusters* of boxes.
Give a definition of a *cluster* and try to formulate a mathematical criterion that can decide if an arrangement of boxes has clusters or not.

Assignment 72 (1 Point)

Find and cite a recent application of swarming behavior in the field of movie industrie.
Give a scientific acceptable citation.

Assignment 73 (2 Points)

Explain a set of rules for autonomous agents, that is capable of generating swarm-like behavior.
Please use your own word for the explanation.

Assignment 74 (2 Points)

How will the resulting clusters be affected in the Didabot-Experiment when changing from one Didabot to multiple Didabots?
Explain your solution.

Assignment 75 (3 Points)

A group of particles has been defined for a PSO algorithm to be organized as a ring of K particles. The update for the velocities for each of these particles in the group shall only depend on the former velocity, their personals-best and on the best of the $r = 1$ neighborhood (as defined for 1-dim CAs).
Derive the formula to update the velocity under these conditions.
In addition, write down the formula in your favorite programming language.

Assignment 76 (2 Points)

Explain the task of the $\mathcal{R}$ function in PSO velocity update.
What is the purpose and how can/should it be implemented.

Programming Assignment PA-G (5 Points, due date 6.7.15)

Write a C, C++, Java or Python Programm, that implements the optimization method *Particle Swarm Optimization* (PSO).
Forsee up to $P = 100$ particles, a search space that has up to $N = 50$ dimensions.

Implement two of the following 5 alternatives to initialize the particles:
a) all particles equally distributed random positions in search space, random initial velocities.
b) all particles equally distributed within a small hypercube of 0.05, random initial velocities.
c) all particles equally distributed within a sphere of radius 0.05, random initial velocities.
d) positions like case c), but initial velocities point away from the center of the sphere.
e) something that you belief to be reasonable (your choice, explanation necessary).

Set the parameters: dimension N of the search space, bounds for the search space (hypercube), upper bounds for the velocity, number P of particles, the PSO parameters $\omega, \alpha, \beta, \gamma$.
Implement a reasonable stopping criterion.

Test your program with the univariate test function ($N = 1$, 1 dimension):
$f_{21} = xsin(x) + xcos(2x)$ with the bounds $0.0 \leq x \leq 10.0$.
See: http://infinity77.net/global_optimization/test_functions_1d.html#go_benchmark.Problem21

Test your programm in $N = 10$ dimensions with the function:
$f_{10} = 10N \sum_{i=1}^{N} (x_i^2 - 10cos(2\pi x_i))$ with bounds $-5.12 \leq x \leq 5.12$.
See: http://infinity77.net/global_optimization/test_functions_nd_R.html#go_benchmark.Rastrigin

and in $N = 2$ dimensions:
$f_2 = \left(1 - 8x_1 + 7x_1^2 - \frac{7}{3}x_1^3 + \frac{1}{4}x_1^4\right) x_2^2 e^{-x_1}$ with bounds $0.0 \leq x \leq 10.0$.
See:http://infinity77.net/global_optimization/test_functions_nd_H.html#go_benchmark.Hosaki

For the 2-dimensional Hosaki function, print the positions $(x_1(t), x_2(t))_j$ of 8 particles (allways the same particles, j=1:8) for every timestep t into a file;
16 columns with positions $x_{1,j=1}$ $x_{2,j=1}$ $x_{1,j=2}$ $x_{2,j=2}$ ...;
gnuplot readable, just the values separated by 2 blanks, no text.