Exam, Natural Computing, Fall 2017

CS version

Prof. Dr. Thomas Bäck

Kaifeng Yang, Hao Wang, Koen van der Blom

January 11th, 2018, 14:00 – 17:00

- Please write readable, and write your name and student identification number on every single sheet of paper for unique identification!
- **Don't spend much time writing long stories – that's not necessary. Don't panic! You ARE able to make it!**
- Look at the whole exam first to get an overview, and do those assignments first which you feel are doable for you.
- Good success and all the best for your studies!

1. **(Natural Computing, total: 6)** To start with, please explain (briefly!) what we mean by complexity and emergence, and give two examples of real-world phenomena illustrating these terms.

2. **(Swarm Intelligence, total: 13)** Many natural computing algorithms rely on the intuition of moving the "population", "swarm", or other solution representations towards better and better solutions as the search proceeds. This requires some kind of selection method. Explain how this selection method works in the following algorithms:

   a. **(2)** Simulated Annealing.
   b. **(2)** Genetic Algorithms.
   c. **(2)** Differential Evolution
   d. **(2)** Particle Swarm Optimization.
   e. **(2)** Ant Colony Optimization.
   f. **(3)** Explain whether you see any fundamental difference in the approach between the first three and the last two methods.

3. **(Evolutionary Algorithms, total: 15)** Assume that you want to solve the 0-1-knapsack problem with an evolutionary algorithm. The problem can be formulated as follows:

   We have n kinds of items, 1 through n. Each kind of item i has a value v_i and a weight w_i. We usually assume that all values and weights are nonnegative. To simplify the representation, we can also assume that the items are listed in increasing order of weight. The maximum weight that we can carry in the knapsack is W. Mathematically the 0-1-knapsack problem can be formulated as:

   Maximize $\sum_{i=1}^{n} v_i x_i$ subject to $\sum_{i=1}^{n} w_i x_i \leqslant W, \qquad x_i \in \{0, 1\}$

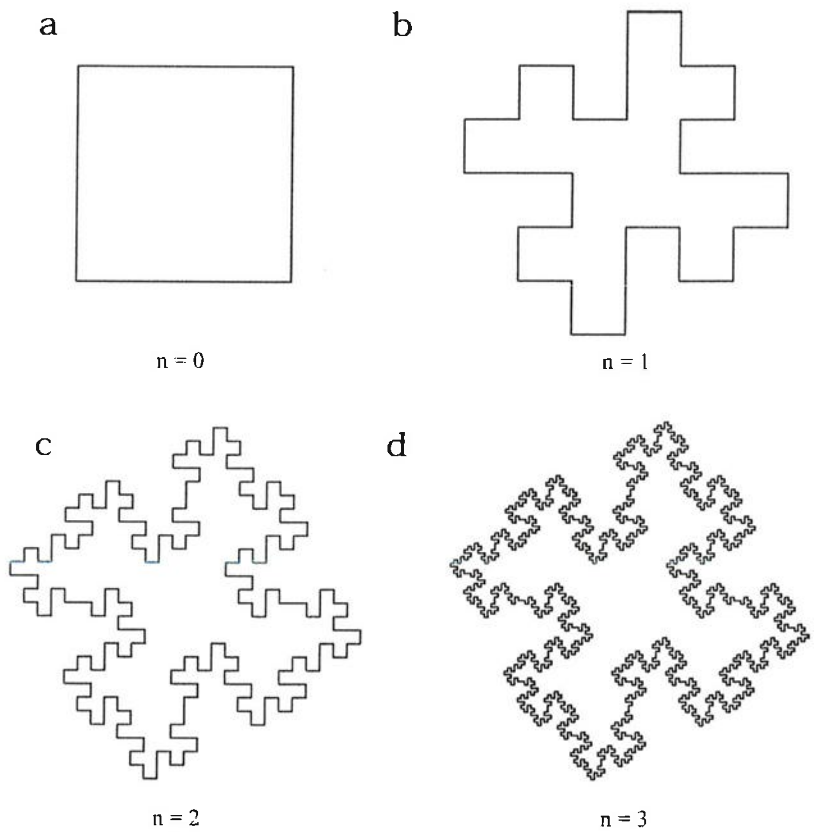
   a. **(9)** Propose a representation and give the mutation and crossover operator for your evolutionary algorithm.
   b. **(3)** Propose a selection operator.
   c. **(3)** Explain how you handle the total weight constraint in your algorithm.

4. **(Simulated Annealing, Evolutionary Algorithms, total: 7)** Assume you want to incorporate concepts from Simulated Annealing into Evolutionary Algorithms. How would you do that? Which concepts would you choose? Why?

5. (**Cellular Automata, total: 15**) Assume you have got a binary 2-D cellular automaton with von Neumann neighbourhood, radius $r = 1$, and toroidal boundaries.

   a. (**3**) How big is the rule space for this type of CA?
   b. (**4**) What does it mean to say something like "rule 6" in this case?
   c. (**4**) For a 3 x 3 CA with initial state: all cells zero except the centre, which is one, simulate the dynamics of rule 6 for one step. Please observe that you need to introduce a neighbourhood indexing convention to do this. Suggestion: leftmost bit = upper cell, next bit = left cell, next bit = middle cell, next bit = lower cell, last bit = right cell.
   d. (**4**) Explain what we mean by "inverse design" of CAs. How would you approach the inverse design problem?

6. **(Lindenmayer Systems, total: 11)** Lindenmayer systems are frequently used to describe the growth of complex structures in biology.
   a. **(6)** Find the axiom and production rule and angle of the Lindenmayer system that generates the following derivations starting from the axiom given in the Figure (a) below.



   b. **(5)** What is the string representation and the turtle graph of the first and second derivation of the following bracketed Lindenmayer system:

(Axiom: F, Production Rule F→ *F[F][-F][+F]*, angle **= 45 degree)**

7. (**DNA Computing, total: 10**) Explain the fundamental idea of DNA computing, three of the operators it is using, and its limitations – if any.

8. (**Neural Computation: 15**) Neural computation is inspired by the biological mechanism of neural cells.

a. **(5)** Perceptron: A perceptron is a simplified neuron receiving inputs as a vector of real numbers, and outputting a real number. Mathematically, a perceptron is represented by the following equation

$$y = f(w_1x_1 + w_2x_2 + \cdots + w_nx_n + b) = f(\vec{w}^T\vec{x} + b)$$

When used for prediction, the model is trained on a set of samples $\{(\vec{x}_1, y_1), \ldots, (\vec{x}_m, y_m)\}$. Any new pattern $\vec{x}_i$ will be assigned the prediction $\hat{y}_i$. In order to optimize the model weights, usually a cost/error function is defined to measure the training error.

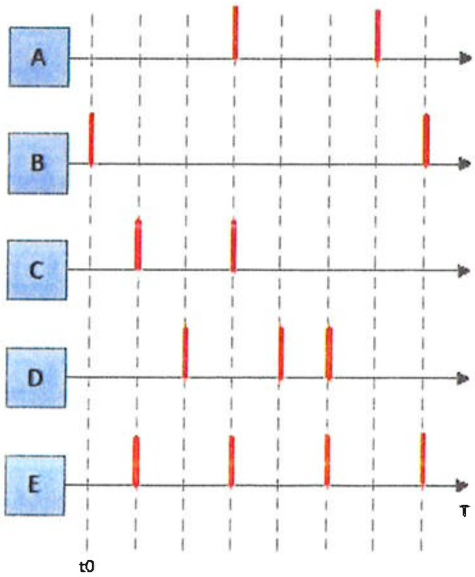
I. **(2)** Please express the **Mean Square Error** cost function mathematically, using the notions given above.

II. **(3)** Normally, the gradient descent method is exploited to optimize the cost function, which requires the gradient computation. Assume the simplest perceptron model (1-d input pattern with linear activation function f):

$$y = f(wx + b), f(x) = ax + c$$

Note that a, b, c are constants. Please differentiate y with respect to w, namely $\frac{dy}{dw}$. Moreover, please differentiate y with respect to x, namely $\frac{dy}{dx}$.

b. **(5)** Neural Coding: an example of five spike trains (ABCDE) is depicted as the following figure, where spikes are drawn as red vertical bars. If we discretize the time as steps (as shown by vertical dashed lines), each spike train can be encoded as an integer. Please encode the information carried by spike train using **spike count** code and **spike latency** (relative to t_0) code. Answer this question in the right table below (hand in this sheet with your name and student id on it or copy the answer to your exam paper).

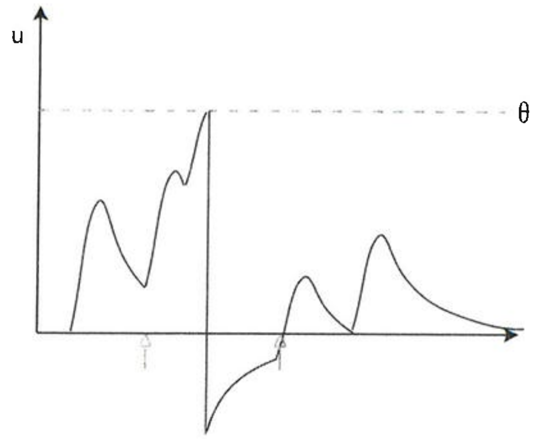


Count	Latency

c. **(5)** Spike Response Model: the basic equation of membrane potential evolution is given as follows:

$$u(t) = \eta\,(t - \hat{t}) + \sum_i w_i \sum_f \epsilon_i(t - t_i^{(f)}),$$

where $\hat{t}$ is the last firing time of the postsynaptic neuron and presynaptic neurons fire at $t_i^{(1)}, t_i^{(2)}, \ldots, t_i^{(f)}$. One spike is generated if the membrane potential reaches the threshold from below. This model can be visualized as the following figure, where the horizontal axis stands for time and the vertical axis represents the membrane potential. In this figure, the firing time of presynaptic neuron are indicated by upward pointing arrows. However, some arrows are missing. Please mark **ALL** the missing presynaptic firing time using arrows **in the figure below**. In addition, please indicate $\hat{t}$ (when the last spike is generated) in the figure blow (the threshold is shown as θ). Hand in the sheet with the figure and write your name and student id on it.



Total: 92 Points possible.