Chalmers University of Technology, Department of Applied Mechanics
Teacher: Mattias Wahde, tel. 772 3727

Exam in FFR 105 (Stochastic optimization algorithms), 2015-10-28, 14.00-18.00, M.
The examiner will visit the exam rooms twice, around 15.00 and around 17.00.

It will be possible to review your results (for the exam and the home problems) during the week starting Nov. 16.

In the exam, it is allowed to use a calculator, as long as it cannot store any text. Furthermore, mathematical tables (such as Beta, Standard Math etc.) are allowed, provided that no notes have been added. However, it is *not* allowed to use the course book, or any lecture notes from the course, during the exam.

Note! In problems involving computation, show *clearly* how you arrived at your answer, i.e. include intermediate steps etc. Only giving the answer will result in zero points on the problem in question.

There are four problems in the exam, and the maximum number of points is 25.

1. (a) *Premature convergence* is a common problem when using genetic algorithms. Define and describe this problem in detail. Next, assuming that (for a particular genetic algorithm) the population size is set to a given, fixed value, and that tournament selection is used, with a fixed tournament selection probability, introduce and describe *two* different ways of preventing premature convergence. (You may describe more than two ways, but you are *not* required to do so: Note that incorrect, additional descriptions may also result in a deduction of points). (3p)

   (b) In genetic algorithms, the concept of *genes* is central. In the biological counterpart, genes serve the purpose of providing the necessary information for generating proteins by means of a process that involves two major steps. Name and *describe* the two steps. (2p)

   (c) Ant colony optimization is based on the cooperative behavior of ants and, in particular, a special form of communication used by ants. Name and describe, in as much detail as possible, this form of communication. (1p)

Continued on the back of this page.

(d) In the Ant system (AS) algorithm, in every iteration, paths are generated probabilistically for each ant. Once the paths have been generated, the pheromone levels are updated. For the case of the standard travelling salesman problem (TSP), describe in detail (with equations and clear descriptions of each equation) exactly how the pheromone levels are updated. Note: You should *only* describe the pheromone update equations, not the entire AS algorithm! (2p)

(e) Gradient descent is a classical optimization method, in which one follows the negative gradient from a given starting point towards a (local) minimum. In this method, starting from a given point $\mathbf{x}_j$ (where $\mathbf{x}$ is a vector and the index enumerates the iterations) one computes iterates such that, once the search direction has been determined, the next iterate $\mathbf{x}_{j+1}$ will depend only on the step length η, so that the *function value* at that point can be expressed as some function $\phi(\eta)$. Consider now the problem of minimizing the function $f(x_1, x_2) = 2x_1^2 + 3x_1x_2 + x_2^2 - 4$ using gradient descent, starting from the point $(x_1, x_2) = (1, 1)$. Find, and write down, the search direction (i.e. a vector with two components) and the expression for the next iterate, inserting numerical values. Then derive (and simplify as much as possible) the expression for $\phi(\eta)$, again with numerical values inserted. Give your answer in the form $a_2\eta^2 + a_1\eta + a_0$, where a_0, a_1, and a_2 are constants. Note: You do *not* have to carry out the line search. It is sufficient that you find the search direction, the next iterate, and $\phi(\eta)$! (2p)

2. In order to study GAs analytically, one often uses functions of unitation, i.e. objective functions in which the fitness depends only on the number (j) of ones in the chromosome.

(a) Consider a simple GA, with a population size of 1, and where the (binary) chromosome is changed using mutations only. The new chromosome is kept if it is better (higher fitness) than the previous one, otherwise it is discarded. Using the Onemax function ($f(j) = j$) as the fitness function, and setting the mutation rate p_{mut} to k/m, where m is the chromosome length, and $k \ll m$ is a positive integer, *derive* an estimate for the runtime of this GA, i.e. the number of iterations required to reach the global optimum. (3p)

(b) In some cases, one makes the further assumption that the population size is infinite. Consider such a case, in which the (binary) chromosomes are initialized randomly, and where a GA with selection only (i.e. no crossover or mutations) is applied to the problem of maximizing the function (of unitation)

$$f(j) = j\left(1 - \frac{j}{m}\right), \tag{1}$$

where m, again, is the chromosome length. Compute

i. The average fitness of the initial population. (1p)

ii. The probability distribution $p_2(j)$ in the second generation (i.e. after one fitness-proportional selection step). (1p)

3. (a) The Lagrange multiplier method depends on a particular relation involving the objective function $f(x_1, x_2, \ldots)$ and the equality constraint function (assuming, here, that there is only one such constraint) $h(x_1, x_2, \ldots)$. Write down this relation (in equation form) and also explain *why* the (local) optima of the objective function, subject to the constraint, occur at the points where this relation holds. You *should* draw a figure (for the case of two dimensions) as a part of your explanation, but you must also describe the figure clearly. (2p)

   (b) Use the Lagrange multiplier method to find the minimum value and the maximum value of the function

$$f(x_1, x_2) = x_1^2 x_2 + 2x_2, \tag{2}$$

   subject to the constraint

$$x_1^2 + x_2^2 - 1 = 0. \tag{3}$$

   (2p)

4. (a) In particle swarm optimization (PSO), there is a specific mechanism for handling the tradeoff between exploration and exploitation. Write down the general equation for the velocity updates in PSO and describe, in detail, the mechanism just mentioned. (2p)

   (b) Consider now a simple one-dimensional application of PSO, in which one is trying to minimize the function $f(x) = (x - \frac{1}{4})^2$, using a swarm size of three. Initially the three particles are located at $x = -1/3$ (particle 1), $x = 0$ (particle 2), and $x = 3/4$ (particle 3), and their speeds are $v = 3$ (particle 1), $v = 1/4$ (particle 2) and $v = -1$ (particle 3) The parameters α and Δt are both equal to 1, w is (here) kept *constant* at the value 1, and $c_1 = c_2 = 2$. Moreover, assume (somewhat unrealistically) that the random numbers q and r are always equal to 1. The initial range $[x_{\min}, x_{\max}]$ is equal to $[-2, 2]$, and the particle speeds are thus restricted to a maximum of 4. Given these parameters, determine, under the PSO algorithm

      i. the velocities and positions of all particles after one iteration (i.e. one updating step for both velocities and positions). (2p)

      ii. the velocities and positions of all particles after two iterations (2p)