Ernst W. Mayr, Rolf Wanka
Manuel Schmitt

Sarntal, 21. September 2014

Project for Course 1, **Moderne Suchmethoden der Informatik: Trends und Potenzial** Ferienakademie 2014

The following exercises should give you the chance to program the optimization approaches from the talks and look how they work. Feel free to choose your favorite programming language. The goal in every exercise is to find a preferably good solution using as few function evaluations as possible. At the end of the course, you are asked to present your results and how you obtained them.

Exercise 1:
Use PSO to minimize the following functions. Note that you have to think of appropriate parameters and of a meaningful behavior for particles that leave the search space. How many iterations and particles do you need for different problem dimensions (e.g., $n = 2, 10, 20, 50, 100$)? The functions are

- the sphere function $\mathit{sphere} : [-100...100]^n \to \mathbb{R}$, where

  $$\mathit{sphere}(\vec{x}) = \sum_{i=1}^{n} x_i^2.$$

  This one is a simple function and mainly a test for your implementation.

- the Rosenbrock function $\mathit{ros} : [-30...30]^n \to \mathbb{R}$, where

  $$\mathit{ros}(\vec{x}) = \sum_{i=1}^{n-1} \left(100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right).$$

  This function has only one local optimum $\mathit{ros}(1, ..., 1) = 0$, which is inside a long valley.

- the Rastrigin function $\mathit{rast} : [-5.12...5.12]^n \to \mathbb{R}$, where

  $$\mathit{rast}(\vec{x}) = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i)\right).$$

  This function has a lot of local extrema, scattert over the whole search space. The global minimum is $\mathit{rast}(0, ..., 0) = 0$.

- the Schwefel function $\mathit{schwefel} : [-500...500]^n \to \mathbb{R}$, where

  $$\mathit{schwefel}(\vec{x}) = \sum_{i=1}^{n} \left(-x_i \cdot \sin\left(\sqrt{|x_i|}\right)\right).$$

  The global optimum, $\mathit{schwefel}(420.9687, ..., 420.9687) = -n \cdot 418.9829$, is close to the corner of the search space and far away from the second best local optimum.

- the Pizza function *pizza* : $[-100...100]^n \to \mathbb{R}$, where

  $$\mathit{pizza}(\vec{x}) = \begin{cases} \sum_{i=1}^n x_i^2, & \exists i,j : x_i \geq 1.1 \cdot x_j \vee x_j \geq 1.1 \cdot x_i \\ 10 \cdot \left(2 \max_{i \neq j} \left\{ \dfrac{x_i}{x_j} \right\} - 2.1\right) \cdot \sum_{i=1}^n x_i^2, & \text{otherwise} \end{cases}$$

  If the coordinates are not too close to each other or if they are negative, this function behaves like the sphere. But along the ray $x_1 = ... = x_n > 0$, the function decreases. The optimum is at the corner, $f(100,...,100) = -10000 \cdot n$.

Exercise 2:
Use Blind Search, Evolutionary Algorithms and maybe PSO to solve the following problems. To do this, you have to think of the definition of the necessary operations, e.g., cross-over and mutation in case of the evolutionary algorithm. Especially reinterpreting the movement equations of the PSO for the discrete situations is not trivial. The problems are:

- The satisfiability problem (SAT). The instances are in the directory `/SAT`. The description of the format can be read under `http://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/satformat.ps`. All formulas are satisfiable.

- The Traveling Salesperson Problem (TSP). The instances are in the directory `/TSP`. The description of the format can be read under `http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/DOC.PS`.

  Note that we appended the explicit length of the edges under EDGE_LENGTH_SECTION, where the triple $i\ j\ k$ means that the distance between node i and node j is k.

- The timetabling problem. The instances are in the directory `/Timetabling`. The description of the problem can be read under `http://www.idsia.ch/Files/ttcomp2002/IC_Problem/node1.html`, the description of the format under `http://www.idsia.ch/Files/ttcomp2002/IC_Problem/node7.html`.