Rolf Wanka
Matthias Kergaßner

Erlangen, 12 May 2022

Exercises to
Swarm Intelligence
Summer 2022
Sheet 3

These exercises will take place on May 17/24 and May 20/27 as a practical (computer) exercise. The room will always be *0.157-115 – CIP Pool EEI*, the times are the regular exercise times. If necessary, further emails regarding this will be sent.

In this exercise, a particle swarm optimizer is to be programmed and tested on some example functions. Choose a programming language yourself (recommendation: Java, C++, Python, or MATLAB).

More information may be obtained from Matthias Kergaßner.

Problem 8:
(a) asks for implementation, (b) for experimantal evaluation. The parts should be considered simultaneously.

(a) First implement a simple PSO (Algorithm 1 on page 6 of the lecture notes) containing only the following components. Here, a, b_{loc} and b_{glob} are scalars:

- Let N denote the number of particles. Choose some N.
- Standard Movement Equations:

$$\begin{aligned} \vec{v}_i^{(k+1)} &= a \cdot \vec{v}_i^{(k)} + b_{loc} \cdot \vec{r}_{loc} \odot (\vec{p}_i^{(k)} - \vec{x}_i^{(k)}) + b_{glob} \cdot \vec{r}_{glob} \odot (\vec{p}_{glob}^{(k)} - \vec{x}_i^{(k)}) \\ \vec{x}_i^{(k+1)} &= \vec{x}_i^{(k)} + \vec{v}_i^{(k+1)} \end{aligned}$$

- $\vec{p}_{glob}^{(k)}$ is the best solution candidate found up to the kth iteration (global attractor), $\vec{p}_i^{(k)}$ is the best solution candidate found up to the kth iteration by particle i (local attractor), $\vec{r}_{loc}$ and $\vec{r}_{glob}$ are random vectors whose components are each uniformly distributed in the interval $[0,1]$. They are chosen freshly in every iteration.
- Initialization of the particles: randomly with uniform distribution in the search space
- Initialization of the velocities: $\vec{v}_i^{(0)} = \vec{0}$
- Minimize the n-dimensional function SPHERE $f : [-100 \ldots 100]^n \to \mathbb{R}$ where $\vec{x} = (x_1, \ldots, x_n)$:

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

  SPHERE is a very simple function with only one local optimum, which, of course, is also the global optimum (hence, it is *unimodal*): $f(0, \ldots, 0) = 0$.
- Use the so-called *Infinity* strategy to cope with the boundaries of the search space: $\forall \vec{x} \notin [-100 \ldots 100]^n : f(\vec{x}) = \infty$.

(b) Try for different dimensionalities (e. g., $n = 2$, $n = 30$, $n = 100$), whether your swarm finds the global optimum. Try different swarm sizes N. Try different settings for the parameters (often used settings: $a = 0.72984$, $b_{loc} = b_{glob} = 1.496172$). What influence of the swarm size, the dimensionality and the choice of parameters do you observe by your experiments?

Problem 9:

Run experiments on the following objective functions (we search for minimum solutions):

- **Rosenbrock:** $f : [-30 \dots 30]^n \to \mathbb{R}$

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

  Rosenbrock has a single optimum, $f(1, \dots, 1) = 0$, which is inside a long, narrow "valley."

- **Rastrigin:** $f : [-5.12 \dots 5.12]^n \to \mathbb{R}$

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

  In contrast to Rosenbrock, Rastrigin has many local optima that are scattered regularly in the search space. The global minimum is $f(0, \dots, 0) = 0$.

- **Schwefel:** $f : [-500 \dots 500]^n \to \mathbb{R}$

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

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

How successful is your PSO implementation on the different functions?

Problem 10:
(optional) Additional to the *Infinity* method, implement and test further methods for bound handling (see Problem 5 on Sheet 2).
Which methods are successful on which of the given functions?

Problem 11:
(optional)

a) Extend your PSO implementations by the two in Problem 7 (Sheet 2) introduced neighborhood patterns: ring and mesh.

b) Compare the neighborhood patterns *complete graph* (that you have used in the implementations above) and *mesh*. Which pattern seems to be best suited on minimizing the 30-dimensional Rosenbrock, and which one for the 30-dimensional Rastrigin? For this, run $1{,}000$ experiments per pattern and function, and compare the means (and the standard deviations) of the best found solution candidates. Try to explain your findings.