Table of content:

Give you an **introduction** on how the idea of pso came up
How the **most basic form** of the algorithm looks and works
What **parameters** we can choose and have to equally **explore** or **exploit** the seearchspace
How the social network looks in terms of **topology**
Where the particle swarm optimization can b**e used** and we also prepared an **implementation** to show
And we will conclude with some **issues** the algorithm still has and lead into a **discussion** with you

I am gonna start right away with a short **video clip** of birds:

Idea comes from **bird flocks searching** for corn. First experiments and simulations conducted almost 30 years ago. Because **no leader** can be found in these flocks. Coordinated turning by individual birds "**voting**" with their bodies on the flight path. **Critical mass** → whole flock turns
Their first algorithm to **simulate bird flocks**:

- Attraction to some place (roost)
- nonlinear attraction to flockmates (do not collide)
- preservation of velocity
- stochastic process which influences the calculation of the next position

Probably also similar to other

- Fish schools: Defence against predators, diluting chance of individual capture, enhanced foraging success and mating, maybe hydrodynamics
- Social behaviour: Humans having more abstract social values, rules, avoid collision, complex interactions, …

PSO has been introduced 1995 by James Kennedy and Russell Eberhard
James Kennedy has since stayed in the field and is also an author of the 2007 paper we present today
Our paper is called Particle Swarm Optimization – an overview, this is what we do today:

The founders of Particle Swarm optimization developed it further to create a powerful optimization method for nonlinear problems.

As they say, the field is in constant movement and researchers discover new ways to do things and new things to do with Pss

Fish schooling: We can see a very suboptimal objective function near the predator

After we heard what the idea of Particle Swarm Optimization is, Lukas will now tell you how a possible basic implementation works:

Optional:
Explanation for bird flocking: This advantage can become decisive, outweighing the disadvantages of competition for food items, whenever the resource is unpedictably distributed in patches” (p.209).
This statement suggests that social sharing of information among conspeciates offers an evolutionary advantage: this hypothesis was fundamental to the developnmt of particle swarm optimization

Open Questions

As we have seen, a PSO has many steps for each of which there are several open questions regarding practical extensions and realizations

Initialization consist of choosing swarm size, particle positions randomly in the search space and random velocities

- Origin seeking bias: As we can imagine on average the tendency of the swarm to find an optimum points closer to the origin of the initialization region
- Hard to find optima outside the initialization region, except from random changes of velocities or position which might heavily depend on parameters, we can only find optima inside of the initialization region
- Premature convergence and slowing-down of the swarm occurs, when the swarm is to one side of the optimum and moves as a coordinated body (groups tend to move slower than individuals)

Other aspects concerning the position or movement update of particles: On which the authors do not know research results yet:

Complete iteration of PSO is a single position update of each particle. But you could also choose particles at random or according to a selection scheme, for example as in parent selection Evolutionary Algorithms: Better performing particles updated more frequently, or worse performing particles get updated more frequently to improve the population as a whole. One could also think about schemes when to update attractor configuration (neighborhood)

You could also explore how particles interact with each other: Implement spherical particles which can not bump into each other, repulsive forces, quantum effects (not further explained)

Not clear how stochasticity of forces and initialization position influences performance

It might happen that particles move outside thee search space or away from the main swarm. How to deal with them? → Reset, moving particle to the edge of the space, …

In most versions a distinction between position of self and neighboring particles is made, which is not necessary. You could also weigh the particles according to their fitness or a ranking

Adaptation

Addition and removal of particles and tuning of PSOs parameters. Goal: Find a completely parameter-free optimizer:

Multitude of swarms with different parameters topologies, sizes and swarms could breed as in an evolutionary algorithm Or parameter tuning between trials But many evaluations needed, so computational effort high. How to do it efficiently, easy and simple?

Theory:

There are many theoretical questions as well. Since we are dealing with an algorithm where many parts interact with each other, it’s hard to prove which parameter are optimal for a particular problem. Although there have been recipes on how to set chi, omega and phi

Or classify a problem such that you can reuse parameters which have been shown to be good for some other problem

Conclusions:

What we have here with PSO, is a simple algorithm that can easily be modified for the specific problems that you want an optimal solution on
It is definitely inspired by nature, although the notion of topologies or neighborhoods having nothing to do with spatial proximity might be a little too abstract for nature.

As we already pointed out: There is still a lot of research to be done. In all the studie sthe problems were not standardized, so Topologies and variants are hard to compare in their applicability

As they say in the paper: PSO is exponentially growing.
So, clearly, we are still looking at a paradigm in its youth, full of potential and fertile with new ideas and new perspectives. Researchers in many countries are experimenting with particle swarms and applying them in real-world applications.
Many more questions have been asked, although still too few have been satisfactorily answered, so the quest goes on.

Click to next slide: Come and join the swarm!