Swarm Intelligence

Lecture Notes

Rolf Wanka

Summer 2022

1 Introduction

Since **Swarm Intelligence** is a large subfield of so-called *Organic Computing* (OC), OC will be introduced in detail at this point. Hence, the following is an introduction to the principles and foundations of the field of organic computing, including a definition of the term *emergence* and an outlook on current research questions and directions.

1.1 Principles of Organic Computing as Superordinate Concept of Swarm Intelligence

Organic Computing refers to computer science in interaction with biology. The following "directions" can be identified:

- **CS** $\rightarrow$ **Bio**: Computer science methods are used to address biological problems: DNA analysis, spatial structures of proteins, ...
- **Bio** $\rightarrow$ **CS**: Use of biological principles in computer science, especially organizing principles for very large *complex* computerized systems.

The lecture deals only with the direction **Bio** $\rightarrow$ **CS**, i.e., we investigate how to use knowledge from biology to cope with *complex* systems. In particular, after this introduction we focus on swarm-based methods.

The complexity of such systems arises mainly from:

(1) • the continuing *miniaturization* of technical systems (IEM)
(2) • the *interconnection* of many "fully equipped" computers
(3) • the *embedding* of hardware and software systems in technical systems such as vehicles or aircraft.

The increasing complexity of technical systems is difficult to control. An error or failure of a (sub-)system generally has effects that cannot be predicted ("what exactly is going on is not always clear", $\rightsquigarrow$ reference to "chaos theory").

Remedies are sought where system development has already been successfully carried out, namely *living systems*. The aim is thus to make knowledge about the functioning of living systems usable for the development of artificial systems.

In the lecture we consider the aspect:

- Application of Swarm Intelligence
  - Swarms
  - Genetic/evolutionary algorithms
  - Search methods

1.2 Self-∗-properties

The correctness of a system refers to the satisfaction of the *safety property* ("bad things never happen") and the *liveness property* ("good things happen sooner or later").

In order to enable the correctness of such a system despite increasing complexity, it becomes more and more important that some processes within the system run automatically, i.e., without human intervention. This is referred to as systems with *self-∗-properties*:

- *self-configuring*
  - A system is self-configuring with respect to a set of actions if it is able to change its own configuration in order to restore or improve the safety property, i.e., to prevent the occurrence of events that are negative for the correctness of the system.

    E.g., adaptation of the formation of a robot swarm depending on the environment, adaptation of the hardware configuration in cloud computing depending on the load, etc.
- *self-optimizing*
  - A system is self-optimizing if – starting from any initial configuration – it is able to optimize the value of a predetermined objective function over the global state. must

    E.g., minimization of energy consumption, ...
- *self-healing*
  - A system is self-healing with respect to a set C of external (bad) events if the occurrence of events from C violates the *safety* property of the system just for a short time. then it heals

    E.g., removing *nodes* from peer-to-peer networks, etc.
- *self-explanatory*
  - The system control explains why it behaved in the observed manner.
- *self-protecting*
  - A system is self-protecting if it continuously maintains the *safety* property in the presence of external actions with "malicious intent".

    E.g., fail-safe in the presence of attacks on peer-to-peer networks, etc.

Advantages of such systems:

- flexible
- robust against failures
- self-optimizing

Disadvantages of such systems:

- can make mistakes
- very long training times, very long times for reconfiguration
- unauthorized interference possible

1.3 Pattern of Organization, Emergence

The following *patterns of organization* are used in such systems:

- *emergence* (see below)
- *autonomy* (state of self-sufficiency, independence, self-government)
- *federation* (cooperation of several (sub-)systems)
- *self-organization*

Emergence is a phenomenon characterized by the *interaction* of many components and the absence of central control and explicitly predetermined patterns. Emergence refers to the formation of structures or properties from the interaction of elements in a complex system. Emergent properties are properties of a whole that cannot be derived directly from the individual parts and can only be explained from the interaction of the parts, i.e., their process (their interaction). The emergent properties also have a feedback effect on the individual components.

Imp

Typical phenomena for emergence are:

- *unpredictability*: Even with perfect knowledge of the individual components, some properties of the overall system cannot be predicted (at first).
- *irreducibility*: The emergent property cannot be derived from the properties of the system components, only the interaction triggers the emergent property: "The whole is more than the sum of the parts".

Examples of emergent behavior:

- oscillating circuit, especially the phenomenon of resonance
- stability of living systems against environmental influence
- performing arts, especially when abstract
- web search (significance of a page is derived from link structure)

1.4 Research Questions/directions

- (better and) quantitative understanding of natural phenomena
- metrics for the assessment of self-organization and emergence phenomena
- system architectures: Observer/Controller architectures
- security: the self-evolution of the OC system must prevent misbehavior and misdevelopment
- inclusion of a-priori knowledge
- cognitive ability (perceptiveness) $\rightsquigarrow$ autonomy and user interaction
- self-explanation

2 Particle Swarm Optimization (PSO)

Swarms of individuals that have to solve a task in a decentralized fashion cooperatively and autonomously have to organize themselves without intervention from the outside. A swarm of individuals can achive by communicating more than just a group of singletons, showing new, emergent capabilities, like two robots that can take stereo pictures, for example. It is a challenging and important scientific task to endow (to equip or supply) swarms of individuals with the so-called self-$*$ properties like self-organization, self-reconfiguration, self-adaptation, self-assembly, self-healing, self-optimization and so on, and to identify and measure the emergent properties.

In the following, we'll present Particle Swarm Optimization (PSO), which has been introduced by Eberhart and Kenndedy in 1995.

- **Given**: a continuous function $F : \mathbb{R}^d \to \mathbb{R}$ (on an interval, if necessary) as a ***black box***. A single function evaluation may possibly be "expensive," i.e., infrequent evaluation is to be aimed at.
- **Goal**: find $\vec{x}_{min} \in \mathbb{R}^d$ such that $F(\vec{x}_{min}) = min(F(\vec{x}) \mid \vec{x} \in \mathbb{R}^d)$

The PSO method works *in an evolutionary manner*, i.e., already obtained solution candidates are improved (step by step). The progress takes place in generations (iterations), where the improvement happens by imitating and learning from other individuals in the population.

A single *particle* (individual) i consists of:

- Position $\vec{x}_i$ (solution candidate)
- Velocity $\vec{v}_i$
- Local best solution $\vec{p}_i$ (*local attractor*).

The *swarm* (population) consists of:

- Particles $1, \ldots, N$.
- Global best position $\vec{p}_{glob}$ (*global attractor*).

$\circledast$

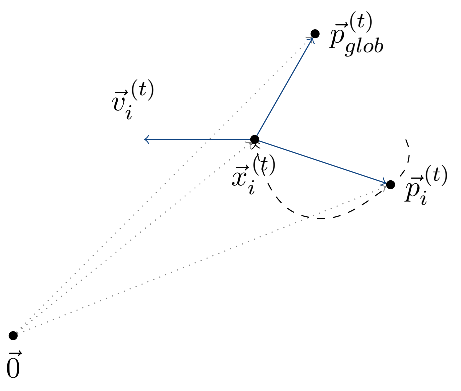

Figure (1) *Situation before applying the plain movement equation to a single particle i. The new velocity $\vec{v}_i^{(t+1)}$ ist the sum of the three vectors leaving $\vec{x}_i^{(t)}$, and the new position $\vec{x}_i^{(t+1)}$ is $\vec{x}_i^{(t)} + \vec{v}_i^{(t+1)}$. ($\vec{0}$ is the origin of the cartesian coordinates)*

$\circledast$?? why

The motion of each particle is influenced by its current velocity, its local attractor, and the global attractor. *In abstract*, this results in the following so-called **movement equations** for the particle i, which describe the position and velocity changes of the particle at the time step $t \rightsquigarrow t+1$.

$$\begin{aligned} \vec{v}_i^{(t+1)} &:= \vec{v}_i^{(t)} + (\vec{p}_{glob}^{(t)} - \vec{x}_i^{(t)}) + (\vec{p}_i^{(t)} - \vec{x}_i^{(t)}) \\ \vec{x}_i^{(t+1)} &:= \vec{x}_i^{(t)} + \vec{v}_i^{(t+1)} \end{aligned}$$

Note: Strictly speaking, the velocity vector still needs to be multiplied by the time unit before it is added to the position vector. This is not necessary here because the time is discretized.

2.1 PSO algorithm

The abstract movement equations above are extended by the so-called *swarm parameters* $a, b_{glob}, b_{loc}, c, d \in \mathbb{R}$, and by "injecting" randomization $\vec{r}_{glob}, \vec{r}_{loc} \in [0,1]^d$, which are random vectors drawn freshly in every iteration.

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

Here, $\odot$ denotes the *Hadamard product* (component-wise multiplication). This leads to the PSO method depicted in Algorithm 1.

For obtaining good results, an interplay between **exploration** and **exploitation** is necessary:

- Exploration: ability of the swarm to search the search space "completely." Important especially at the starting phase of the algorithm.
- Exploitation: Ability of the swarm to "find near good solutions even better solutions" (idea: "if here is a good solution, there could be an even better solution in the vincinity"). This is important especially towards the ending phase of the search.

Algorithmus 1 particle swarm optimization (PSO)

```
input: $f : \mathbb{R}^d \to \mathbb{R}$, $N$: Number of particles
output: $\vec{p}_{glob} \in \mathbb{R}^d$. $\mathbb{R}^d$.
  {initialization of particles}
  for $i := 1$ to $N$ do
    initialize $\vec{x}_i$ and $\vec{v}_i$ randomly
    initialize $\vec{p}_i := \vec{x}_i$
  end for$\vec{p}_i$
  {initialize global attractor}
  initialize $\vec{p}_{glob} := argmin(f(\vec{p}_i) \mid i = 1, \ldots, N)$
  {simulation of swarm behavior}
  repeat
    {simulation of individual particles}
    for $i := 1$ to $N$ do
      $\vec{v}_i := a \cdot \vec{v}_i + \cdot b_{glob} \cdot \vec{r}_{glob} \cdot (\vec{p}_{glob} - \vec{x}_i) + b_{loc} \cdot \vec{r}_{loc} \odot (\vec{p}_i - \vec{x}_i)$
      $\vec{x}_i := c \cdot \vec{x}_i + d \cdot \vec{v}_i$
    end for
    {update local and global attractors}
    for $i := 1$ to $N$ do
      if $f(\vec{x}_i) \leq f(\vec{p}_i)$ then
        $\vec{p}_i := \vec{x}_i$
      end if
      if $f(\vec{x}_i) \leq f(\vec{p}_{glob})$ then
        $\vec{p}_{glob} := \vec{x}_i$
      end if
    end for
  until termination criterion reached
```

2.2 Convergence Criteria

if it runs for infinite time

The objective is that the swarm **converges**, i.e., that the particles of the swarm "agree" on a single solution. However, note that this does not necessarily have to be the actual optimum of the objective function!
The swarm parameters turn out to be very important to achive convergence. Therefore we want to determine **how the parameters of the swarm have to be chosen in order to actually achieve convergence**.

order is importance

We assume that the swarm converges to a point $\vec{p}$. And, in case of convergence, the individual particles are not allowed to move anymore, i.e., $\vec{v}_i \rightarrow 0$.

automatically approximately

The vector components in the movement equations are decoupled[1] (*) ^"after infinite many iterations," therefore a 1-dimensional analysis is sufficient. (in infinite case)
For an arbitrary but fixed particle as well as for an arbitrary but fixed dimension, according to the movement equations at time k, the following applies: (after decoupling)

$$v^{(k+1)} = a \cdot v^{(k)} + b_{glob} \cdot r_{glob} \cdot (p_{glob}^{(k)} - x^{(k)}) + b_{loc} \cdot r_{loc} \cdot (p_{loc}^{(k)} - x^{(k)})$$
$$x^{(k+1)} = c \cdot x^{(k)} + d \cdot \underbrace{v^{(k+1)}}_{\text{new velocity}}$$

simple product as Decoupled

local influence

We set $r_{glob} = r_{loc} = \frac{1}{2}$ for the analysis, i.e., to the expected value of the two random variables. Furthermore, we substitute:

↳ for $k \to \infty$ case

$$b := \frac{1}{2} \cdot (b_{loc} + b_{glob})$$
$$p^{(k)} := \left(\frac{b_{loc}}{b_{loc} + b_{glob}}\right) \cdot p_{loc}^{(k)} + \left(\frac{b_{glob}}{b_{loc} + b_{glob}}\right) \cdot p_{glob}^{(k)}$$

percentage of local attractors

This yields the following simplified movement equations:

$$v^{(k+1)} = a \cdot v^{(k)} + b \cdot (p^{(k)} - x^{(k)}) \quad (1)$$
$$x^{(k+1)} = c \cdot x^{(k)} + d \cdot v^{(k+1)} \quad (2)$$

coupled linear sys
x depend on v
v " " x

Equation 2 can now be solved for $v^{(k+1)}$. Moreover, a calculation for $v^{(k)}$ can be given from it by calculating "one time step backward":

$$v^{(k+1)} = \frac{1}{d} \cdot (x^{(k+1)} - c \cdot x^{(k)}) \quad (3)$$
$$\Rightarrow v^{(k)} = \frac{1}{d} \cdot (x^{(k)} - c \cdot x^{(k-1)}) \quad (4)$$

→ which we can sub in Eq ①.

Now the new Equations 3 and 4 can be substituted into the Equation 1 so that after multiplying both sides by d the following equation is obtained:

$$x^{(k+1)} - c \cdot x^{(k)} = a \cdot (x^{(k)} - c \cdot x^{(k-1)}) + b \cdot d \cdot (p^{(k)} - x^{(k)})$$

(only x var).

By rearranging, we obtain:

$$x^{(k+1)} + (\underline{b \cdot d} - c - a) \cdot x^{(k)} + a \cdot c \cdot x^{(k-1)} = \underline{b \cdot d} \cdot p^{(k)} \quad (5)$$

(We get a recursion in x)

[1] the individual dimensions are independent of each other, i.e., in this case each dimension can be considered separately from the others

In Equation 5, we note that the two swarm parameters b and d only appear together as a product. For this reason,

$$d = 1 \quad (\text{as } b \neq 1)$$

can be fixed and b determined accordingly. The swarm parameter d can thus be omitted, since the same effect can be achieved by changing the value of b accordingly.

In $K \to \infty$ case → If $p^{(k)}$ would not change any more. i.e. would be a fixed point, the $x^{(k)}$ should move towards it due to the assumed convergence, i.e., there must be a fixed point X for the Equation 5. Thus for this fixed point holds:

→ $x^{k+1} = x^{k}$ (new point same as old)

$$X + (b - a - c) \cdot X + a \cdot c \cdot X = b \cdot p^{(k)} \tag{6}$$
$$\Leftrightarrow (1 + b - a - c + a \cdot c) \cdot X = b \cdot p^{(k)} \tag{7}$$

After the particle approaches the point $p^{(k)}$, $X \to p^{(k)}$ holds and so, according to Equation 7:

$$1 - a - c + a \cdot c \overset{!}{=} 0 \tag{8}$$
$$\Leftrightarrow (a - 1)(c - 1) \overset{!}{=} 0 \tag{9}$$

To satisfy Equation 9, we set

$$c = 1\,,$$

i.e., in addition to d, another swarm parameter vanishes.
Substituting $c = d = 1$ into the Equations 1 and 2 yields the following two equations:

$$v^{(k+1)} = a \cdot v^{(k)} + b \cdot (p^{(k)} - x^{(k)})$$
$$x^{(k+1)} = (1 - b) \cdot x^{(k)} + \underbrace{a \cdot v^{(k)} + b \cdot p^{(k)}}_{\to v^{(k+1)}}$$

This can be written as:

$$\underbrace{\begin{pmatrix} x^{(k+1)} \\ v^{(k+1)} \end{pmatrix}}_{=:Y^{(k+1)}} = \underbrace{\begin{pmatrix} 1-b & a \\ -b & a \end{pmatrix}}_{=:A} \cdot \underbrace{\begin{pmatrix} x^{(k)} \\ v^{(k)} \end{pmatrix}}_{=:Y^{(k)}} + \underbrace{\begin{pmatrix} b \\ b \end{pmatrix}}_{=:B} \cdot \overbrace{p^{(k)}}^{\to p} \tag{10}$$

← called disturbance

$$\Leftrightarrow Y^{(k+1)} = A \cdot \underbrace{Y^{(k)}}_{\text{old pair}} + B \cdot p \quad \Leftrightarrow Y^{(k+1)} = A^{(?)} Y^{(0)} + A^{(?)} Y^{(1)} + \ldots \tag{11}$$

One can show that this linear system of equations has the equilibrium $\begin{pmatrix} p \\ 0 \end{pmatrix}$ and that the system converges if the magnitudes of the eigenvalues λ_1, λ_2 (they may be complex numbers!) of A are strictly smaller than 1. For the eigenvalue determination we calculate the determinant of $(A - \lambda \cdot I)$ and set it equal to 0:

↓ so that the higher power of λ, and in turn of the matrix A decreases to convergence

$$det \begin{pmatrix} 1-b-\lambda & a \\ -b & a-\lambda \end{pmatrix} = (1 - b - \lambda) \cdot (a - \lambda) + a \cdot b \overset{!}{=} 0 \tag{12}$$

$$\Rightarrow \underbrace{||\lambda_{1,2}||}_{\to \text{length of complex number}} = ||\frac{a-b+1}{2} \pm \sqrt{\frac{(a-b+1)^2}{4} - a}|| \overset{!}{<} 1 \tag{13}$$

So now we ask how the swarm parameters a and b must be chosen so that the Equation 13 is satisfied, i.e., that $||\lambda_{1,2}|| < 1$. This is the case if $a < 1$, $b > 0$ and $2a - b + 2 > 0$. Figure 2 represents the combinations of values of a and b that satisfy this condition.

$$\lambda^t = \begin{pmatrix} d_1^t & & 0 \\ & d_2^t & \\ 0 & & \ddots \; d_n^t \end{pmatrix}$$

$a > 0$ $\quad$ $(a+1)b - 4a > 0$ $\quad$ $2ab - 2b > 0$

$$A = (D \lambda D^{-1})^2$$
$$D \lambda \underbrace{D D^{-1}}_{1} \lambda D^{-1}$$

if $t \to \infty$ $\lambda^t = 0$

$\frac{(a-b+1)}{4}$

$(a+1)^2 + b^2 - 2$

$a^2 - 2a + 1 + b^2 -$

$= D\lambda^2 D$

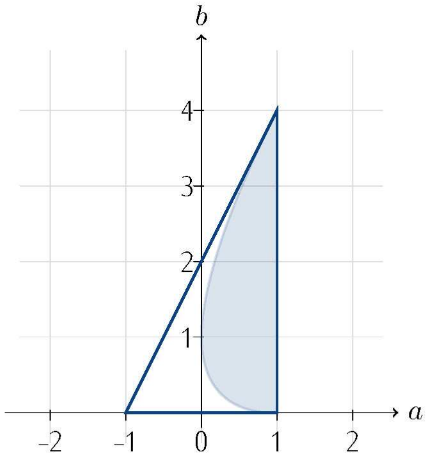

Figure (2) *For parameter values of a and b that lie within the drawn triangle, the system converges. Within (approximately) the lightly backed area, harmonic oscillation occurs because the eigenvalues in this case are complex.*

Depending on the actual eigenvalues of the matrix A, the oscillation behavior of the system varies. Figure 3 presents different behavior types. It is taken from

> I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters* 85 (2003) 317–325. doi:10.1016/S0020-0190(02)00447-7

Complex parts of λ, sum up to zero, So, they converge

one λ is complex

one λ is -ve So which causes oscillation

just +ve λ_1, λ_2

-ve λ_i

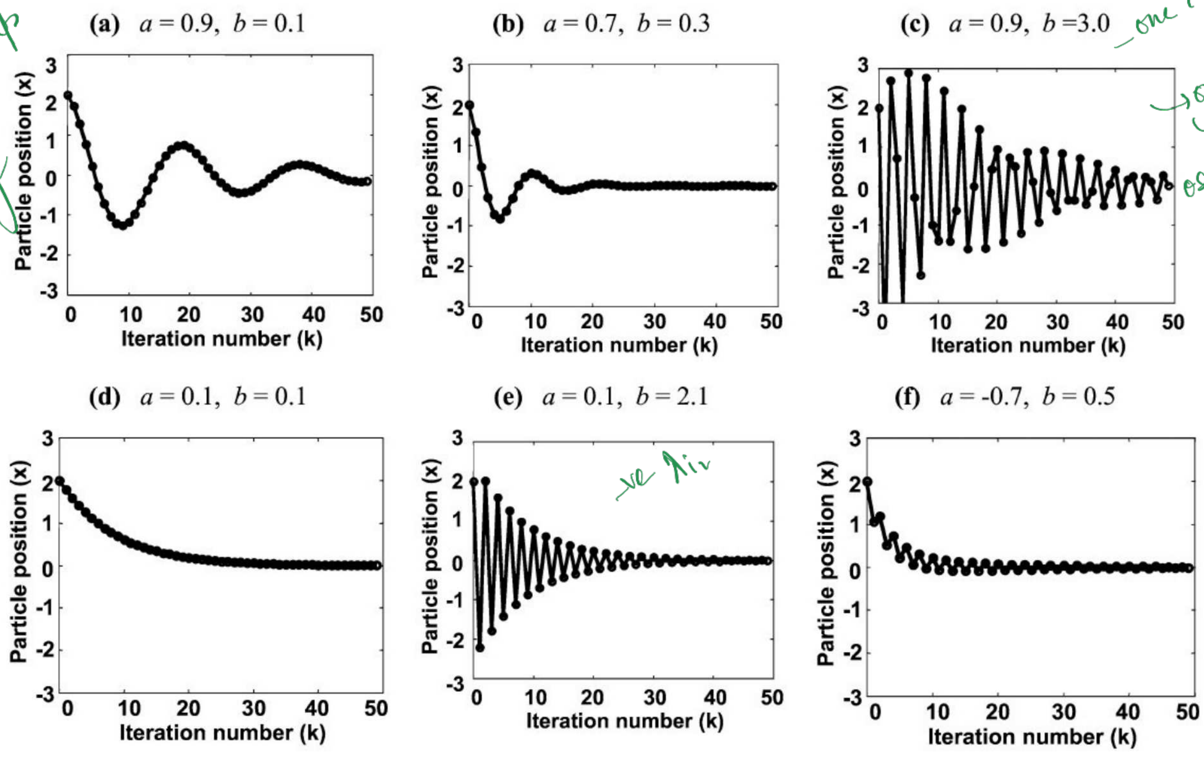

Fig. 3. Examples of dynamic behavior of a single particle for various choices of the parameters a and b. (a) Harmonic oscillations with slow convergence. (b) Harmonic oscillations with quick convergence. (c) Harmonic oscillations with zigzagging. (d) Non-oscillatory convergence. (e) Symmetric zigzagging. (f) Asymmetric zigzagging.

Bratton and Kennedy, experts for applications, suggest $a = 0.72984$, and $b_{loc} = b_{glob} = 1.49617$.