Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Matematica

Advanced Particle Swarm Optimization

Relatore: Prof. Marco Scianna

Candidato: Davide de Vergori
Matricola: S331571

Anno Accademico 2024–2025

Contents

1 The Standard Particle Swarm Optimization Model **3**

1.1 Optimization Framework 3

1.2 Canonical Version of the PSO 3

1.3 Motivation of the work 7

2 A Modified PSO for Biological Systems **8**

2.1 Reformulated Dynamics 8

3 Numerical Settings **10**

3.1 Parameters 11

3.2 Results 12

3.3 Varying model coefficients 13

3.4 Discussion 23

3.5 Future works 23

to my ker*:*

alla Caia, alla Lussy, a Tonino

Abstract

In recent years, with the advent of Artificial Intelligence, the field of optimization has deeply attracted the attention of the scientific community. In this context, Particle Swarm Optimization technique has been developed for the training of neural networks. Recently, Particle Swarm Optimization algorithm has been used also in biology, where the cell has the role of the agent of the swarm. A promising variant of the algorithm makes use of the bioinspired concept of neighbohrood, where a cell can interact only with a small number of cells around her.

Introduction

Particle Swarm Optimization (PSO) is an optimization algorithm belonging to the family of meta-heuristic. Due to his simplicity and relatively low computational cost, it has been used in various engineering fields and applications, spanning from weight training of neural networks, combinatorial optimization problem and travel sales problem.

It has been firstly introduced by Eberhart and Kennedy in 1995 [1]. and the basic idea is to introduce a set of agents, called particles or swarm. The particles moves in the mathematical domain (or search space), sharing information each other, looking for the optimum of the optimization problem.

In classical PSO, every particle is influenced by his personal best position and the best position of the swarm. In this way, the dynamic of the swarm evolves via communication and interaction in more promising regions of the search space, trying to avoid regions characterized by local minima.

Despite the use of PSO algorithm in classical engineering fields, PSO applications in biology is a recent and promising line of research. In the last years, with the born of the so called quantitative biology, classical and advanced mathematical instruments have been used to predict complex and highly non linear phenomena, such as reconstruction of metabolic networks, structure protein prediction and genetic sequences. In this context, PSO algorithm has shown great ability to find approximate solutions, with a reasonable balance between exploration of the search space and exploitation of acquired information.

Moreover, PSO algorithm could be easily hybridized with other classical stochastic optimization techniques, such as genetic algorithms, simulated annealing or clustering, giving the possibility to adopt a specific procedure for a specific problem. This could be relevant especially in biological context, where complexity of data and models require flexibility between accuracy of results and computational constraints.

This work is organized as follow. The first chapter introduce a comprensive description of the mathematical formulation of the model and of the optimization algorithm. The second chapter describe an alternative formulation of the problem, based on the concept of neighbohrood. The third chapter shows simulations of the proposed model, both for unimodal and multimodal test function, and a critical analysis of the results.

Chapter 1

The Standard Particle Swarm Optimization Model

Particle Swarm Optimization (PSO) is an algorithm designed to solve optimization problems by simulating the collective behavior of a group of simple agents, referred to as particles. These agents explore a defined search space in pursuit of the optimal solution to a given objective function. Each particle evaluates its current position and adjusts its movement based on both its own experience and the collective knowledge of the group.

1.1 Optimization Framework

Let $F(\mathbf{x})$ be a real-valued objective function defined over a d-dimensional domain:

$$F(\mathbf{x})\colon \mathbf{X} \subseteq \mathbb{R}^d \to \mathbb{R}$$

The goal is to identify $\mathbf{x}^* \in \mathbf{X}$ that minimizes $F(\mathbf{x})$, i.e.,

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathbf{X}} F(\mathbf{x}) \subseteq \mathbb{R}^d.$$

the domain $\mathbf{X}$ represents the search space, and each candidate $\mathbf{x} \in \mathbf{X}$ corresponds to a possible solution. In other words, a solution $\mathbf{x}^*$ belongs to $\mathbf{X}^*$ if and only if it minimizes the objective function $F(\mathbf{x})$. For simplicity, we hereafter suppose that the search space $\mathbf{X}$, which also represents the domain of $F(\mathbf{x})$, is a parallelepiped (box) in the d-dimensional real space, i.e., $\mathbf{X} \subseteq \mathbb{R}^d$, with $d \in \{1, 2, 3\}$.

1.2 Canonical Version of the PSO

In the canonical version of the PSO, the exploratory swarm is set to be formed by N point particles. Each agent $i = 1, \ldots, N$ is characterized by a given state $\mathbf{s}_i(t)$, being $t \in T \subset \mathbb{N}_+ \cup \{0\}$ the discretized time variable (i.e., the number of iterations of the algorithm): it essentially contains its present position in the search space $\mathbf{x}_i(t) \in \mathbf{X}$, its

speed $\mathbf{v}_i(t) \in \mathbb{R}^d$, and a memory vector $\mathbf{p}_i(t) \in \mathbf{X}$, that stores the best (i.e., in terms of optimization/minimization of the objective function) domain location so-far visited by the particle of interest:

$$\mathbf{s}_i(t) = (\mathbf{x}_i(t), \mathbf{v}_i(t), \mathbf{p}_i(t)) \in \mathbf{X} \times \mathbb{R}^d \times \mathbf{X}.$$

The sequence of states acquired over time by the i-th agent gives its history and the sequence of positions gives its trajectory.

At each iteration of the algorithm, the vector $\mathbf{p}_i(t)$ is updated with the following rule:

$$\mathbf{p}_i(t+1) = \begin{cases} \mathbf{p}_i(t), & \text{if } F(\mathbf{x}_i(t+1)) > F(\mathbf{p}_i(t)); \\ \\ \mathbf{x}_i(t+1), & \text{if } F(\mathbf{x}_i(t+1)) \leq F(\mathbf{p}_i(t)), \end{cases} \tag{1.1}$$

with initial value $\mathbf{p}_i(0) = \mathbf{x}_i(0)$, for any particle $i = 1, \ldots, N$. In other words, the actual best position of a given agent has to be intended as the point(s) of its own trajectory corresponding to the so-far found minimal value of the cost function, i.e., without any influence of its groupmate performance/dynamics. In this respect, we have the following chain of relations:

$$F(\mathbf{p}_i(0)) \geq F(\mathbf{p}_i(1)) \geq F(\mathbf{p}_i(2)) \geq \ldots \geq F(\mathbf{p}_i(t)),$$

for any time step $t \in T$. The above rule (1.1) enforces the exploratory potential of the population, as the memory vector is updated even if there is not an effective improvement of the minimization solution.

In classical PSOs, the position update of any given particle is set to depend on (i) its past movement (*inertia/persistence*), (ii) its entire history (the *cognitive* component), (iii) the information transmitted by other groupmates (the *social* component), with biases given by random contributions. In particular, the dynamics of the representative i-th agent are defined by the following rules, i.e., for any $t \in T$:

$$\overline{\mathbf{x}}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1); \tag{1.2a}$$

$$\mathbf{v}_i(t+1) = \underbrace{m\mathbf{v}_i(t)}_{\mathbf{v}_i^{\text{in}}(t+1)\text{: inertia}} + \underbrace{c_1\mathbf{R}_1(t+1)(\mathbf{p}_i(t) - \mathbf{x}_i(t))}_{\mathbf{v}_i^{\text{cogn}}(t+1)\text{: cognitive component}} + \underbrace{c_2\mathbf{R}_2(t+1)(\mathbf{p}_{\text{g}}(t) - \mathbf{x}_i(t))}_{\mathbf{v}_i^{\text{soc}}(t+1)\text{: social component}}, \tag{1.2b}$$

and

$$\mathbf{x}_i(t+1) = \begin{cases} \overline{\mathbf{x}}_i(t+1), & \text{if } \overline{\mathbf{x}}_i(t+1) \in \mathbf{X}; \\ \\ \overline{\partial\mathbf{X}}(\overline{\mathbf{x}}_i(t+1)), & \text{otherwise,} \end{cases} \tag{1.3}$$

where $\overline{\partial \mathbf{X}}(\overline{\mathbf{x}}_i(t+1))$ is a boundary operator that maps $\overline{\mathbf{x}}_i(t+1) \notin \mathbf{X}$ to a given point in $\mathbf{X}$. For instance, it may project $\overline{\mathbf{x}}_i(t+1)$ to the nearest point on the border of $\mathbf{X}$ or implement absorbing/reflecting walls. Invisible walls, that allow particles to fly out from the search space and be no longer considered may be also set to replace (1.3).

The vector $\mathbf{p}_g$ stores the *global* best location, in terms of minimization of the objective function, found so-far by the entire swarm, that is derived by comparison of the best solutions actually obtained by the component agents, i.e., for any $t \in T$

$$\mathbf{p}_g(t) = \underset{i=1,\dots,N}{\arg\min}\{F(\mathbf{p}_i(t))\}, \tag{1.4}$$

starting form

$$\mathbf{p}_g(0) = \underset{i=1,\dots,N}{\arg\min}\{F(\mathbf{p}_i(0))\} = \underset{i=1,\dots,N}{\arg\min}\{F(\mathbf{x}_i(0))\}. \tag{1.5}$$

Random diagonal matrices $\mathbf{R}_1(t)$ and $\mathbf{R}_2(t) \in \mathbb{R}^{d\times d}$ contain independent entries uniformly sampled from $[0,1]$, generated for any agent and iteration:

$$(\mathbf{R}_{1,i})_{jj} = (\mathbf{R}_{2,i})_{jj} \sim \mathcal{U}([0,1]), \quad \forall j = 1,\dots,d,\ t \in T, i = 1,\dots,N.$$

Some remarks:

- In the context of Newtonian mechanics, if we define:

  $$\mathbf{f}_i(t+1) := \mathbf{v}_i^{\text{cogn}}(t+1) + \mathbf{v}_i^{\text{soc}}(t+1), \text{ for any } t \in T \text{ and } i = 1,\dots,N,$$

  We can think at $\mathbf{f}_i$ as the resulting external force acting on the i-th particle, while the actual change of i-th particle velocity can be written as:

  $$\Delta\mathbf{v}_i(t+1) = \mathbf{v}_i(t+1) - \mathbf{v}_i(t) = \mathbf{f}_i(t+1) - (1-m)\mathbf{v}_i(t).$$

  In this fashion, the evolution of the system can be seen as a dynamic equilibrium between forces. The coefficient m can be interpreted as an *inertia weight*, whose inclusion has been shown greatly improve the algorithm performance. The quantity $(1-m)$ can also be seen as a friction coefficient, where high values of m correspond to particle moves in a very low viscosity medium. On the other hand, lower values of m imply a more dissipative system, with an high probability for the system to stagnate into a local optima. Some authors have also proposed a decrease in the value of m, both linear [2] and non linear trend [3].

- The parameters c_1 and c_2 can be interpreted as *acceleration coefficients* that establish how much the exploring particles are affected by the cognitive attractor and the social attractor, respectively. In general, these two coefficients, often referred to as cognitive and social learning factors, respectively, are set to be equal, constant in time, and shared between all agents. Also in this case, various alternative have been introduced: Carlisle and Dozier [4] have proposed to differentiate both values, in particular setting $c_1 = 2.8$ and $c_2 = 1.3$. The good performance of this choices has been further confirmed by Schutte and Groenwold [5]. Also time-varying learning factors have been introduced, For instance, in Ratnaweera et al. [6] they have been both set to decrease in time, whereas in Yin et al. [7] the authors have proposed to decrease c_1 and simultaneously increase c_2.

- in classical PSO, for what concern the social component, there is a transmission of information across the *entire* swam, as each particle knows the best position so-far found by *all* its groupmates. Such a version of the method, usually referred to as *global* PSO, has been shown to have high converge speed but also a large probability to stagnate into sub-optimal solutions.

  To overcome this issue, local variants of the algorithms have been proposed, i.e., characterized by the fact that any particle is set to communicate with only a subset of groupmates. In this respect, any agent $i = 1, \ldots, N$ has its own best solution vector, that depends on the information it receives by the particles it is actually communicating with, i.e.,

$$\mathbf{p}_{\mathrm{g},i}(t) = \underset{j \in N_i(t)}{\arg\min}\{F(\mathbf{p}_j(t))\}, \tag{1.6}$$

  where $\mathcal{N}_i(t)$ is the set of agents that at any time $t \in T$ share knowledge with i. The vector $\mathbf{p}_{\mathrm{g},i}(t)$ has then to replace its global counterpart $\mathbf{p}_{\mathrm{g}}(t)$ in the velocity update rule of the representative agent i, being also to be included in its state, i.e.,

$$\mathbf{s}_i(t) = (\mathbf{x}_i(t), \mathbf{v}_i(t), \mathbf{p}_i(t), \mathbf{p}_{\mathrm{g},i}(t)) \in \mathbf{X} \times \mathbb{R}^d \times \mathbf{X} \times \mathbf{X}.$$

  In local PSOs, the swarm is indeed not entirely attracted towards the single point $\mathbf{p}_{\mathrm{g}}$: rather, multiple social attractors can emerge thereby increasing differentiation within the population (and therefore its exploratory potential). As a consequence, the convergence speed is reduced but the chance of finding global optima is increased.

  The neighborhood of the i-th agent, regardless its metric or topological nature, is defined *open* when it excludes the particle i itself and *closed* when i itself is included: in this second case, we have that

$$F(\mathbf{p}_{\mathrm{g},i}(0)) \geq F(\mathbf{p}_{\mathrm{g},i}(1)) \geq F(\mathbf{p}_{\mathrm{g},i}(2)) \geq \ldots \geq F(\mathbf{p}_{\mathrm{g},i}(t)),$$

  for any $i = 1, \ldots, N$ and time step $t \in T$. Social networks topology can be classified into two main families: static and dynamic ones. In the former, the groups of communicating particles do not change over time, i.e., $\mathcal{N}_i(t) = \mathcal{N}_i$ for any agent i and time t. They are then denoted as symmetric if the relationship between a pair of particles is same for each one. To give some examples, Eberhart and Kennedy [1] have used the simple and static ring communication structure, where each individual is connected to two adjacent members, with a toroidal wrapping. In the latter, an adaptive time-varying topology connectivity can be performed [8] or it can be changed over time the number of agents with which any given particle can share information [9].

  On the other hand, the global PSO can be conceptualized as a local PSO where all agents have a closed neighborhood which includes, at any time, the entire swarm. In this respect, in [10] it has been suggested that the use of local neighborhoods seems better for early exploration of the search space whereas a global point of view helps to increase convergence in the later phases of the optimization process. In this respect, these authors have proposed to increase over time the number of agents with which a particle can communicate, until having a fully connected swarm.

1.3 Motivation of the work

- In classical PSO, the whole population share information with all other particles: this is not realistic for biological cells, where information is only shared with an appropriate neighborhood and instantaneous information is forbidden.

This work tries to give an answer to this point with some numerical experiments.

Chapter 2

A Modified PSO for Biological Systems

To align PSO with biological principles, we introduce a new formulation suitable for two-dimensional domains. This model separates the computation of movement direction from speed, neglect inertia weight and introduces stochastic motion as a distinct influence.

2.1 Reformulated Dynamics

Let $F(\mathbf{x})\colon \mathbf{X} \subseteq \mathbb{R}^2 \to \mathbb{R}$ represent the objective function. For each particle, we define the new velocity as follows:

$$\mathbf{v}_i(t+1) = v_i\, \hat{\mathbf{w}}_i(t+1) \tag{2.1}$$

where v_i is a scalar coefficient representing motility (accounting possible physiological limitations) and $\hat{\mathbf{w}}_i(t+1)$ is a unit vector. In the following, vectors with $\hat{}$ has to be intended as unit vectors, i.e. $\widehat{\mathbf{a}} = \dfrac{\mathbf{a}}{|\mathbf{a}|}$, where $\mathbf{a} \in \mathbb{R}^{\mathbf{2}}$. The two quantities have different biophysical meanings. For instance, cell speed is related to the concept of cell *motility*, established by the frequency of retraction/expansion cycles of plasmamembrane (PM) motility structures, such as filopodia and pseudopodia, which is in turn highly controlled by intracellular cascades involving specific ions and molecules (e.g., calcium, Rac1, Rho). The direction of movement of a cell is instead dictated by the spatial organization of its cytoskeletal filaments. They can either randomly arrange or align in response to external inputs (due, e.g., to the presence of other individuals or to specific environmental conditions), thereby establishing a preferred orientation of the individual body and movement.

Moreover, $\mathbf{w}_i(t+1)$ is computed as:

$$\mathbf{w}_i(t+1) = \alpha\Big(\mathbf{p}_i(t) - \mathbf{x}_i(t)\Big) + \beta\Big(\mathbf{p}_{\mathrm{g},i}(t) - \mathbf{x}_i(t)\Big) + \gamma\, \hat{\mathbf{r}}_i(t+1). \tag{2.2}$$

The effective velocity of the exploratory agents is fact a trade-off between personal and global information, with a bias given by random effects, which are now decoupled from

the other migratory traits. Here:

- $\mathbf{p}_i(t)$ is the particle's best-known position defined as before.
- $\mathbf{p}_{g,i}(t)$ is the global best position defined for every particle $i = 1, \ldots, N$ with respect to his neighborhood, i.e.:

$$\mathbf{p}_{g,i}(t) = \underset{j \in N_i(t) \cup \{i\}}{\arg\min} \{ F(\mathbf{x}_j(t)) \}, \tag{2.3}$$

  being $N_i(t)$ the neighborhood of a particle, defined as:

$$N_i(t) = \{i\} \cup \{ j = 1, \ldots, N,\ j \neq i : \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \leq 2d_{adh} \}, \tag{2.4}$$

  and d_{adh} being the maximal extension that can be reached by the plasma-membrane adhesive structure of each of the two interacting cells.

- $\hat{\mathbf{r}}_i(t+1)$ represents stochastic term of the velocity defined as:

$$\hat{\mathbf{r}}_i(t+1) = \begin{pmatrix} \cos\Big(\eta_i(t+1)\Big) \\ \sin\Big(\eta_i(t+1)\Big) \end{pmatrix} \tag{2.5}$$

  where $\eta_i(t+1)$ is a stochastic variable uniformly distributed over the interval $[0, 2\pi)$.It is a noise term that takes into account that biological elements (not only cells but also bacteria and other organisms) crawl in a *random* fashion to explore the surrounding environment. A wide range of sophisticated or application-related laws may be used: however, we implement simple Brownian (diffusive) movements.

- α, β, γ are non-negative coefficients satisfying $\alpha+\beta+\gamma = 1$. The term with coefficient α models mesenchymal behavior (individual, self-driven migration). The second term models epithelial migration, relying on collective behavior and long-range signaling. The random term captures intrinsic cellular noise.

Chapter 3

Numerical Settings

To test our numerical implementation, two objective functions will be used: in particular, the first one is an unimodal function, i.e., a function with only a global minimum (Fig. 3.1):

$$F_1 = x^2 + y^2$$

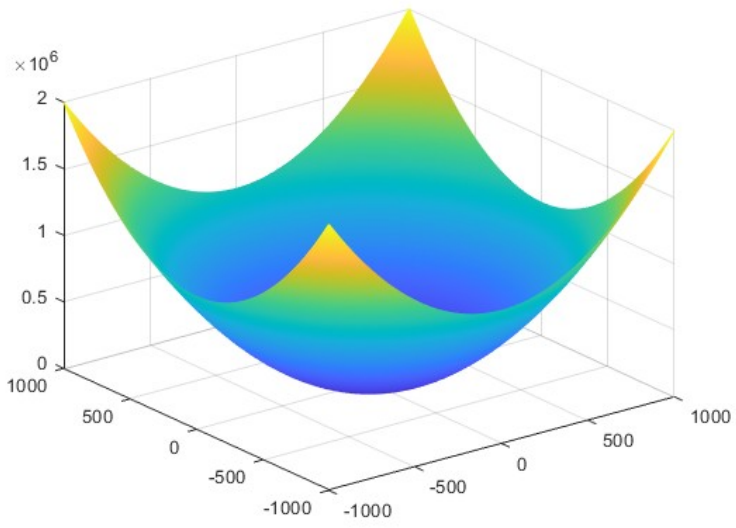

Figure 3.1: Test Function F_1

while the second one is a multimodal function, i.e., a function with one global minimum and several local minima. Among many possible choices, we choose the so called Rastrigin Function (Fig. 3.2):

$$F_2 = 20 + x^2 + y^2 - 10\Big(\cos(2\pi x) + \cos(2\pi y)\Big).$$

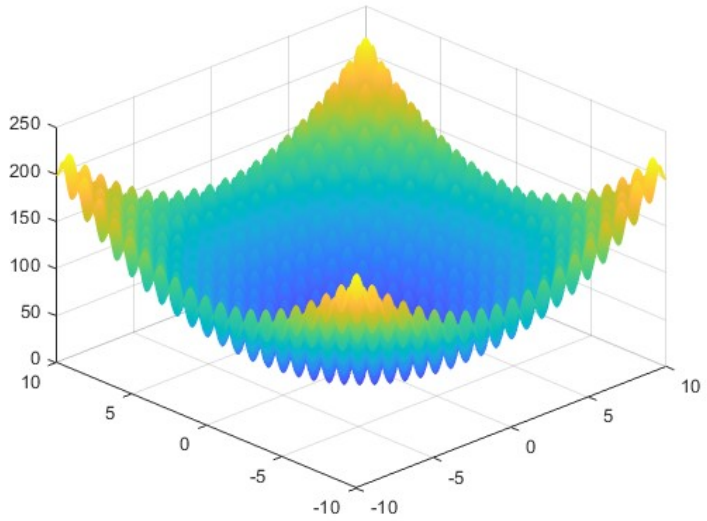

Figure 3.2: Test Function F_2

Test Function F_2 has been plotted in the $[-10, 10] \times [-10, 10]$ domain to appreciate local minima.

Unimodal functions are problems that any optimization technique should be able to solve with a good degree of resolution: they are typically used to identify good local optimizers and the best parameter setting of the tested algorithm. On the other hand, multimodal functions represent challenging issues for optimization algorithm: they may both be unable to distinguish among the promising regions and settle on a single optimum or be trapped into one or more optimal solution.

Both of them will be evaluated in the closed domain

$$X = ([-1000, 1000] \times [-1000, 1000])\,\mu\text{m}^2,$$

corresponding to a square of $1\,\text{mm}^2$, having both global minimum in $(0, 0)$.

3.1 Parameters

Cell population size N is set equal to 100 for any simulation, a good compromise between the computational cost and the exploratory potential of the population.

We set $d_{adh} = 60$ μm for the radius of interaction between cells, as observed in many experimental work. This parameter plays a crucial role since it establish the interactions between cells and the consequent formation of the neighbohrood.

Moreover, individual speed v_i is set equal to 1 μm/sec.

Boundary conditions are chosen of the Mirror Walls type [11]: every time a particle is at the boundary of the domain, the sign of the velocity is inverted. In other words, the particle is reflected in the domain.

Final observation time t_f is set equal to 5000 iterations, a value sufficiently high to establish the algorithm behaviour, i.e., a sufficiently high value to permit to every cell to explore the whole domain.

Initialization of the position of the particles is established according to the uniform distribution, while initial velocity of particles is set to zero. Moreover, cells move according to laws (1.2a), (1.3), (2.1), (2.2).

Coefficients α, β, γ are the same values for every particle, constant in time and chosen s.t. $\alpha + \beta + \gamma = 1$.

3.2 Results

We choose 15 different combinations of parameters for both the test functions. For any combination, we perform $R = 25$ runs, i.e., 25 simulations resulting different because of the stochastic nature of the algorithm (initial positions of the particle randomly chosen at every run and a stochastic noise term on the velocity of the particle).

Two quantities are chosen to classify the output, both of them in percentage:

- **Success Rate (SR).** It measures the fraction of runs (among a total amount of R) in which the cell population finds the optimal solution of the problem at least with one of its members:
$$SR = \frac{1}{R} \sum_{k=1}^{R} S_k,$$
where S_k is defined as:
$$S_k = \mathbf{1}\Big(\min_{i=1,\dots,N} \|p_i(t_f) - x^*\|^2 < \delta_{\text{succ}}^2 \Big)$$
and $\mathbf{1}(z)$ is the indicator function (1 if z is true; 0 if z is false), t_f is the final time and $\delta_{\text{succ}} = 10^{-3}$. SR can be seen as a measure of the ability of the algorithm to achieve the solution with a fixed error.

- **Collective Convergence Rate to the exact solution (CCR):**
$$CCR = \frac{1}{R} \sum_{k=1}^{R} CC_k,$$
where CC_k is defined as:
$$CC_k = \mathbf{1}\Big(\max_{i=1,\dots,N} \|p_i(t_f) - x^*\|^2 < \delta_{\text{succ}}^2 \Big),$$
i.e., a quantity which establishes if the optimal solution of the problem is found by the entire group of particles.

In this respect, four possible scenario can be distinguished:

- S_1, when $SR = CCR = 0$. It is the worst situation, since it implies that the algorithm has no success in every run and no agent is able to find the solution.
- S_2, when $0 < SR \leq 50\%$, regardless the value of CCR. In this case, in the $SR - \%$ of the runs, at least one particle has been able to solve the optimization problem.
- S_3, when $50\% < SR \leq 100\%$, regardless the value of CCR. Also in this case, in the $SR - \%$ of the runs, at least one particle has been able to solve the optimization problem.
- S_4, when $SR = CCR = 100\%$. It is the best situation, as the algorithm is able to reach the minimum point in every run with all the agents.

3.3 Varying model coefficients

In this section, we analyze how coefficients variations influence the behaviour of the algorithm in terms of solving the optimization problem. We use the simplex graph for visualization of results, a tetrahedron with every edge equal to one, typical tool in multidimensional data analysis. Two graphs are shown, one in the 3D space and one in the $\gamma = 0$ plane.

In the following, red points are used for Scenario S_1, yellow points for S_2, blue points for S_3 and finally green points for S_4.

Test Function

$$F_1 = x^2 + y^2$$

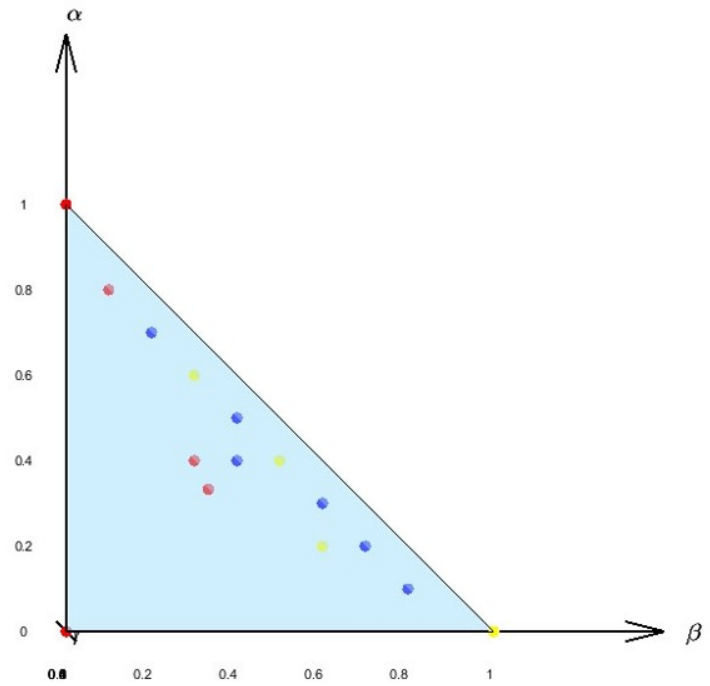

Figure 3.3: Simplex of results in the $\gamma = 0$ plane

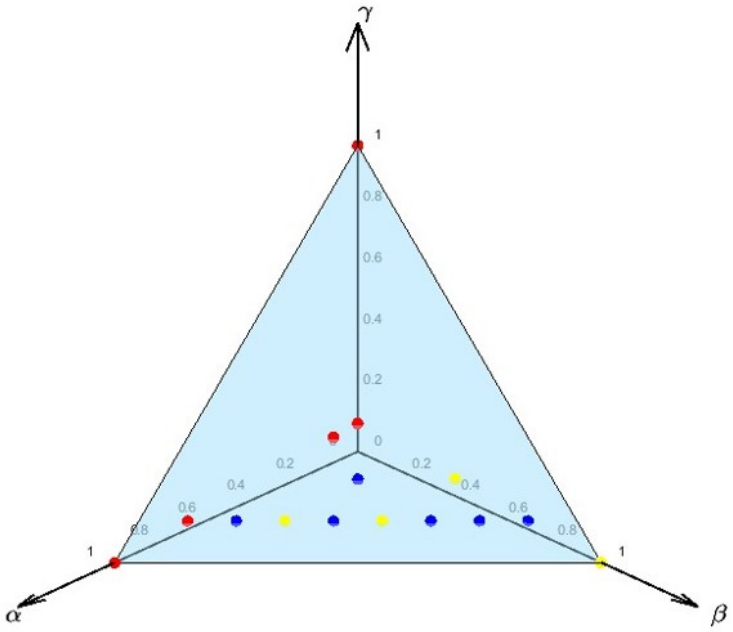

Figure 3.4: 3-D Simplex results

For all chosen combinations of parameters, scenario S_4 is never obtained in any simulation. Furthermore, high values of γ (i.e. $\gamma \in [0.3, 1]$) always result in scenario S_1, i.e., never solving the minimization problem.

On the other hand, low values of γ (i.e. $\gamma \in [0, 0.2]$) and therefore intermediate and high values of α and β, always let the algorithm solve the optimization problem (Fig. 3.3 and 3.4).

Very good results (i.e. scenario S_3) are obtained for values of $\beta \in [0.4, 0.8]$.

In the following, time-lapse sequence of particle dynamics at iteration $t = 0$, $t = 2500$ and $t = 5000$ respectively are shown, for every possible Scenario (S_2 and S_3 lead us to very similar time-lapse for a fixed set of parameters).

Scenario S_1 $(\alpha = \frac{1}{3},\ \beta = \frac{1}{3},\ \gamma = \frac{1}{3})$

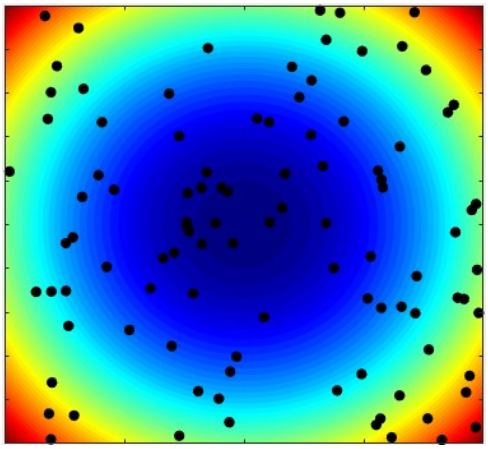

Figure 3.5: Scenario S_1, $N = 0$

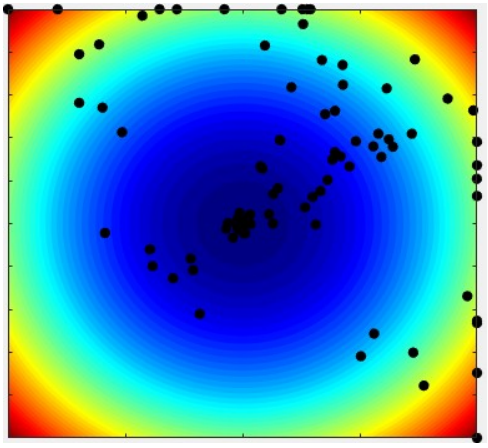

Figure 3.6: Scenario S_1, $N = 2500$

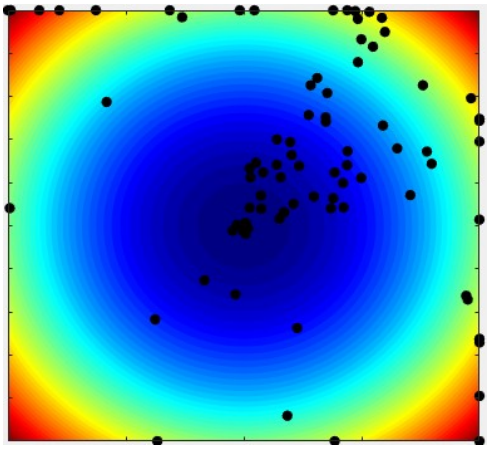

Figure 3.7: Scenario S_1, $N = 5000$

Scenario S_3 $(\alpha = 0.4,\ \beta = 0.4,\ \gamma = 0.2)$

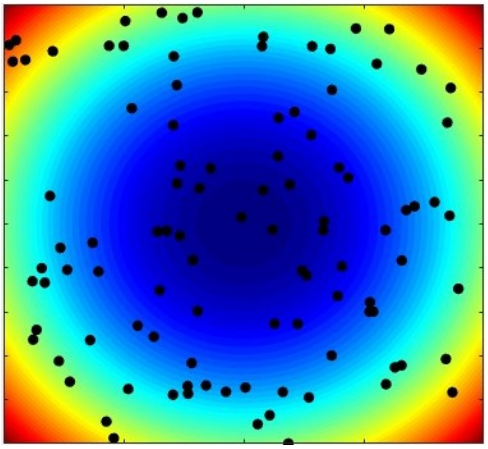

Figure 3.8: Scenario S_3, $N = 0$

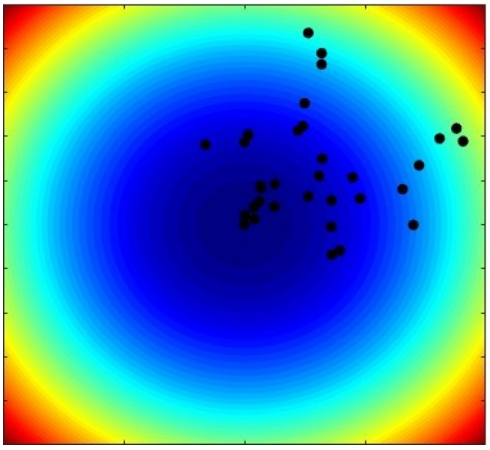

Figure 3.9: Scenario S_3, $N = 2500$

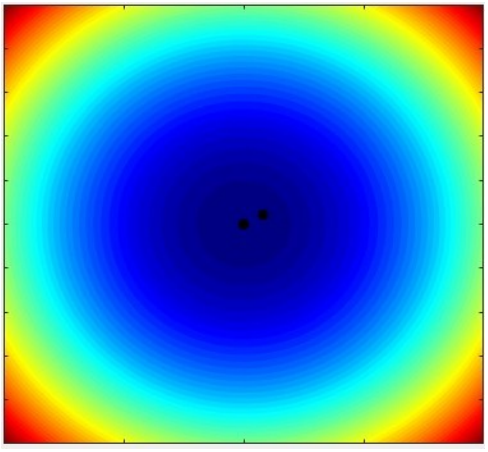

Figure 3.10: Scenario S_3, $N = 5000$

Test Function

$$F_2 = 20 + x^2 + y^2 - 10\big(\cos(2\pi x) + \cos(2\pi y)\big).$$

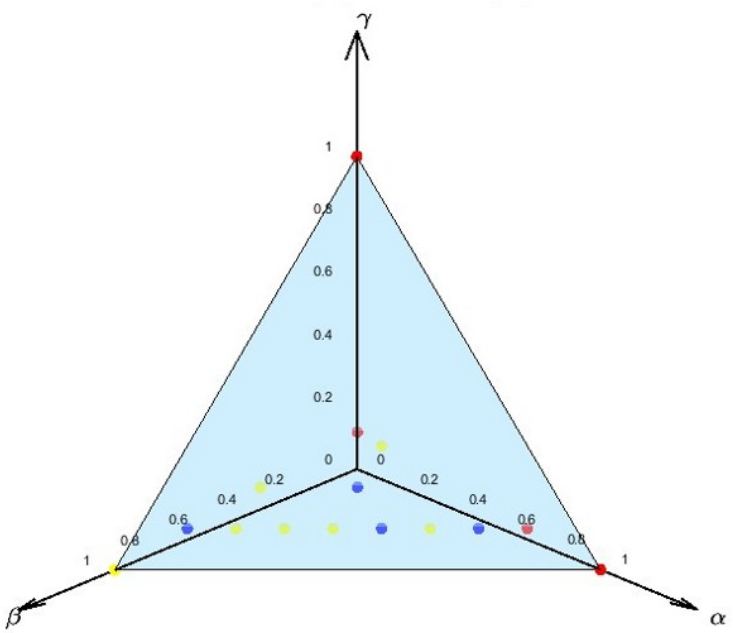

Figure 3.11: 3-D Simplex results

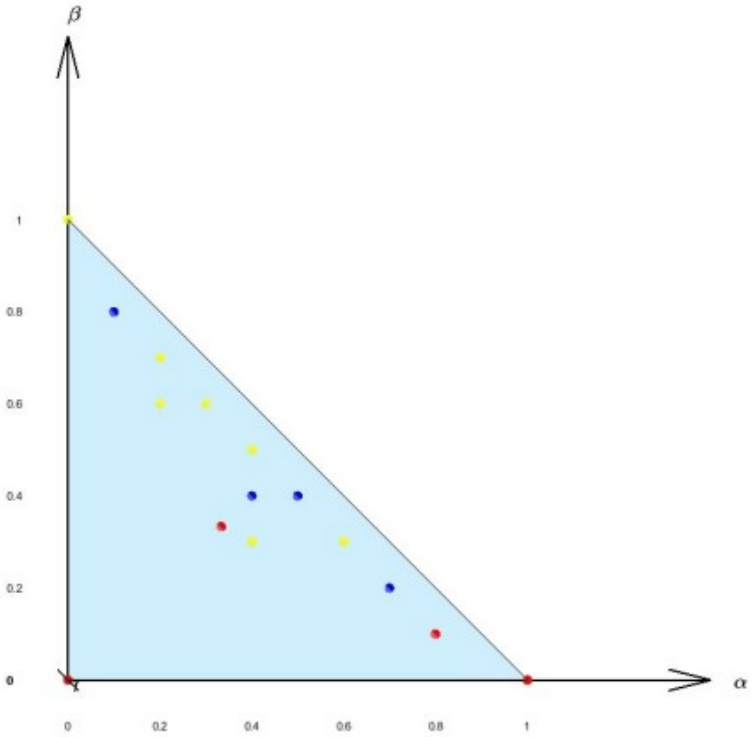

Figure 3.12: Simplex of results in the $\gamma = 0$ plane

As for function F_1, scenario S_4 is never obtained in any simulation. Furthermore, high values of γ (i.e. $\gamma \in [0.3, 1]$) always result in scenario S_1, as for function F_1.

On the other hand, low values of γ (i.e. $\gamma \in [0.1, 0.3]$) and therefore intermediate and high values of α and β, always let the algorithm solve the optimization problem, i.e. Scenario S_2 and S_3.

No specific values of α and β are needed to achieve a good result, i.e. $SR \geq 88\%$.

As for function F_1, time-lapse sequence of particle dynamics at iteration $t = 0$, $t = 2500$ and $t = 5000$ respectively are shown, for every possible scenario.

Scenario S_1 $(\alpha = \frac{1}{3},\ \beta = \frac{1}{3},\ \gamma = \frac{1}{3})$

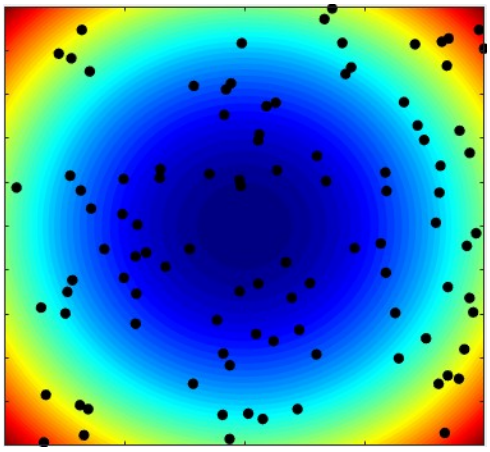

Figure 3.13: Scenario S_3, $N = 0$

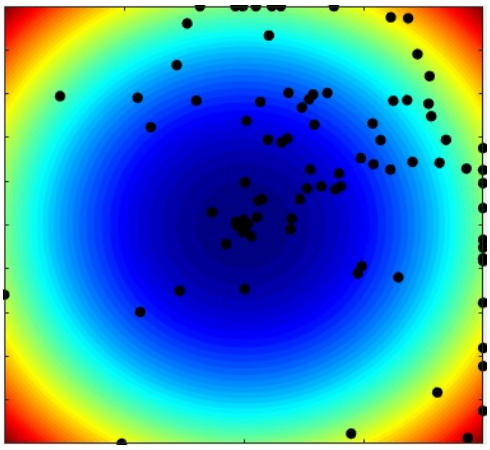

Figure 3.14: Scenario S_3, $N = 2500$

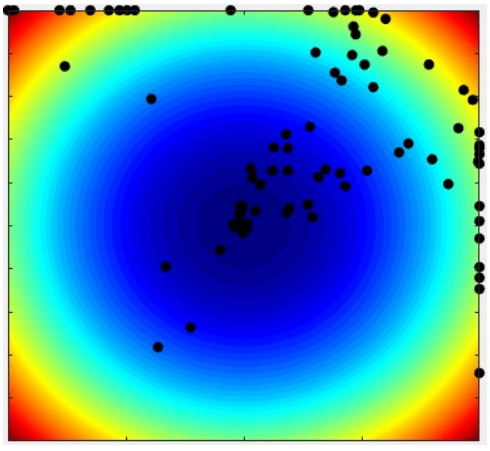

Figure 3.15: Scenario S_3, $N = 5000$

Scenario S_3 $(\alpha = 0.7,\ \beta = 0.2,\ \gamma = 0.1)$

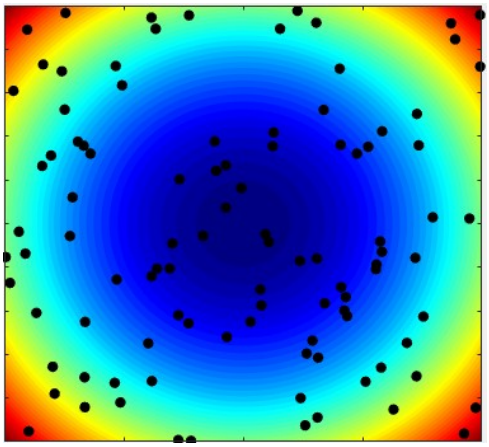

Figure 3.16: Scenario S_3, $N = 0$

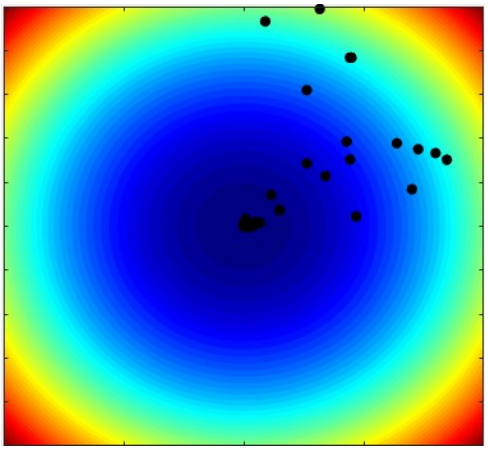

Figure 3.17: Scenario S_3, $N = 2500$

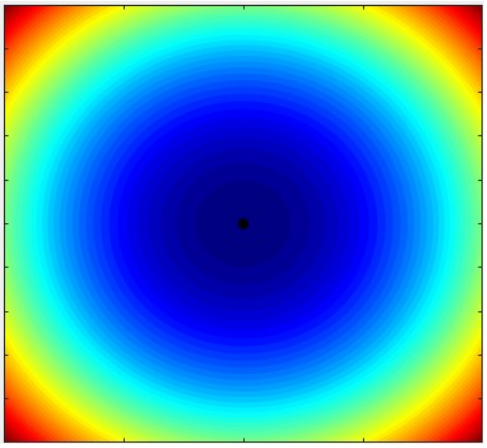

Figure 3.18: Scenario S_3, $N = 5000$

Comparison

For both unimodal and multimodal function and for every combination of the parameters, the whole population is never able to reach the minimum point at t_f (i.e. Scenario S_4).

This is probably due to the chosen parameter for the radius of interaction between cells: 60 μm could be a very small value with respect to size of the domain. From a biological perspective, this means that every cell is not able to create an appropriate neighbohrood s.t. the information, i.e. the exploration of the whole domain, could be easily shared between cells. In other words, in this way, our model become too much "local".

Furthermore, (except for the tern $\alpha = 0.4,\ \beta = 0.3,\ \gamma = 0.3$ for F_2 Test Function) values of $\gamma > 0.3$ never lead us to Scenario S_2 or S_3, cause probably the particle "lose herself" in the domain due to the random noise. This is also probably due to the size of the domain with respect to the radius of interaction. In fact, if the particle at $t = 0$ is far away from the minimum, high weight for the stochastic term of the velocity never let the particle find an appropriate descent direction.

On the other hand, the quality of the solution relies on the values of α and β. Scenario S_3 comes out with a lower percentage for function F_2 with respect to function F_1. This is due to the smoothness of function F_1 with respect to function F_2.

There's no evident distinction in terms of quality of the solution for the social component with respect to the cognitive component (Fig 3.4, 3.12): we have good results both for $\alpha \in [0.1, 0.8]$ and for $\beta \in [0.1, 0.8]$.

In confirmation of this, the tern $\alpha = 0.4,\ \beta = 0.4,\ \gamma = 0.2$ lead us to $SR = 100\%$ and $SR = 88\%$ for functions F_1 and F_2, respectively.

3.4 Discussion

As highlighted in [12] and [13], converge property of PSO is always a weak equilibrium between parameters values, i.e. between exploration and exploitation.

In fact, high values of γ with respect to α and β, could result in a dynamic of Brownian confined crawling, which often do not lead to convergence. This is well known in biology, where population of cells show collective phenomena (chemotaxis) only when the deterministic part of the signal is higher than the random one. In fact, even if a low value of γ could be useful for exploration, collective phenomena comes out only when the deterministic part of the signal is higher than the random noise. Moreover, also in cell migration and animal flocking has been shown [14] [15] that if social forces are lower than random noise, this could lead to disordered dynamics.

These considerations confirm the obtained results (scenario S_1 and S_2) when high values of γ are chosen.

On the other hand, the deterministic part of the parameters, α and β, help reaching the minimum, guiding cells in an appropriate descent direction. Our results are also in agreement with respect to the "self-looping" and "swarm collapse" phenomena [16] : the former comes out with high values of α, where the particle is inclined to explore alone the whole domain, while the latter comes out with high values of β, where a set of particle is inclined to reach a sub-optimal region of the domain. In particular, this is true for F_2 test function. These considerations suggests that a good balance between alpha and beta values could lead to better results. This is true for our simulations, where the tern $\alpha = 0.4,\ \beta = 0.4,\ \gamma = 0.2$ lead to scenario S_3 for F_1 test function.

3.5 Future works

Future directions of this modeling approach could include:

- Introduce a persistence term in cell movement instead of the random term: this could help the cell in following a promising descent direction.
- Introduce differentiation in the population, i.e. splitting population into two families, leader and follower: the former move only according to their personal best while the latter follow the leader within an appropriate neighborhood. This correspond to a biological scenario where leader cells release a chemical factor and follower cells move by chemotaxis.

Bibliography

[1] James Kennedy and Russell Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95 - International Conference on Neural Networks* 4 (1995), pp. 1942–1948.

[2] Yuhui Shi and Russell C. Eberhart. "A modified particle swarm optimizer". In: *1998 IEEE International Conference on Evolutionary Computation* (1998), pp. 69–73.

[3] Yong Lu, Jie Wang, and Tao Li. "Biological network reconstruction using particle swarm optimization". In: *BMC Systems Biology* 9.1 (2015), p. 24.

[4] A. Carlisle and G. Dozier. "An off-the-shelf PSO". In: Jan. 2001.

[5] Jaco Schutte and Albert Groenwold. "A Study of Global Optimization Using Particle Swarms". In: *Journal of Global Optimization* 31 (Jan. 2005), pp. 93–108. DOI: 10.1007/s10898-003-6454-x.

[6] Asanga Ratnaweera, Saman Halgamuge, and Harry Watson. "Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients". In: *Evolutionary Computation, IEEE Transactions on* 8 (July 2004), pp. 240–255. DOI: 10.1109/TEVC.2004.826071.

[7] Shiyuan Yin et al. "Reinforcement-learning-based parameter adaptation method for particle swarm optimization". In: *Complex Intelligent Systems* 9 (Mar. 2023), pp. 5585–5609. DOI: 10.1007/s40747-023-01012-8.

[8] Wei Hong Lim and Nor Ashidi Mat Isa. "Particle swarm optimization with increasing topology connectivity". In: *Engineering Applications of Artificial Intelligence* 27 (2014), pp. 80–102. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2013.09.011. URL: https://www.sciencedirect.com/science/article/pii/S0952197613001851.

[9] Xiaohui Hu and Russell C. Eberhart. "Adaptive particle swarm optimization". In: *Proceedings of the 2002 Congress on Evolutionary Computation* 2 (2002), pp. 1470–1474.

[10] P.N. Suganthan. "Hierarchical overlapped SOM's for pattern classification". In: *IEEE Transactions on Neural Networks* 10.1 (1999), pp. 193–196. DOI: 10.1109/72.737507.

[11] Jacob Robinson and Yahya Rahmat-Samii. "Rahmat-Samii, Y.: Particle Swarm Optimization in Electromagnetics. IEEE Trans. on Antennas and Propagation 52(2), 397-407". In: *Antennas and Propagation, IEEE Transactions on* 52 (Mar. 2004), pp. 397–407. DOI: 10.1109/TAP.2004.823969.

[12] Maurice Clerc. "Particle swarm optimization". In: *Wiley Encyclopedia of Operations Research and Management Science* (2010).

[13] Ioan Cristian Trelea. “The particle swarm optimization algorithm: convergence analysis and parameter selection”. In: *Information Processing Letters* 85.6 (2003), pp. 317–325.

[14] Iain D. Couzin et al. “Effective leadership and decision-making in animal groups on the move”. In: *Nature* 433.7025 (2005), pp. 513–516.

[15] Tamás Vicsek et al. “Novel type of phase transition in a system of self-driven particles”. In: *Physical Review Letters* 75.6 (1995), p. 1226.

[16] Frans Van den Bergh and Andries P. Engelbrecht. “Cooperative particle swarm optimizer with spatial neighborhood topology”. In: *IEEE Transactions on Evolutionary Computation* 8.3 (2004), pp. 225–239.