Field of study: Master of Science Management and Data Science

Report on module Complementary Study

Optimization Methods

submitted by
Priet-Nguyen, Huyen (3041484)

Lüneburg, 15.02.23

Examiner: Prof. Dr. Paolo Mercorelli

Contents

1 Model of Lotka-Volterra for Three Species

1.1 The model

Recall the report [2], the underlying Lotka-Volterra Three Species Prey - Predator - Top Predator in Specific Case of Sheep - Wolf - Shepherd is written in the following system. where $x(t)$ is population of sheeps, $y(t)$ is population of wolfs, $z(t)$ is that of shepherds. g_i are the growth rates of that population, d_{ij} are the interaction gain of the population i when meeting j, where $i, j \in \{x, y, z\}$.

$$
\begin{aligned}
\frac{dx}{dt} &= g_x x(t) + d_{xy} x(t) y(t) + d_{xz} x(t) z(t) &= \dot{x} \\
\frac{dy}{dt} &= g_y y(t) + d_{yx} x(t) y(t) + d_{yz} y(t) z(t) &= \dot{y} \\
\frac{dz}{dt} &= g_z z(t) + d_{zx} x(t) z(t) + d_{zy} y(t) z(t) &= \dot{z},
\end{aligned} \tag{1.1}
$$

1.2 Assumptions of sheep-wolf-shepherd

The following assumptions are considered:

1. Start with the general three species prey (sheep) - predator (wolf) - top predator (shepherd) model.

2. When **sheep** meet **wolf**: sheep population will decrease, and wolf population will grow.

3. When **sheep** meet **shepherd**: sheep population will grow, and shepherd population will grow. (shepherd will cultivate sheep, while at the same time use them for its growth)

4. When **wolf** meet **shepherd**: they kill each others (with more wolf killed than shepherd).

Above asumptions lead to the rules as below.

$$
\begin{array}{llll}
x: & d_{xy} < 0 & d_{xz} > 0 & \\
y: & d_{yx} > 0 & d_{yz} < 0 & d_{yz} > d_{zy} \\
z: & d_{zx} > 0 & d_{zy} < 0 & d_{zy} < d_{yz}
\end{array} \tag{1.2}
$$

The population equilibrium are calculated by following equations.

$$
\begin{aligned}
Px_{eq} &= \frac{g_x d_{yz} d_{zy} - g_y d_{xz} d_{zy} - g_z d_{xy} d_{yz}}{d_{xy} d_{yz} d_{zx} + d_{xz} d_{yx} d_{zy}} \\
Py_{eq} &= \frac{-g_x d_{yz} d_{zx} + g_y d_{xz} d_{zx} - g_z d_{xz} d_{yx}}{d_{xy} d_{yz} d_{zx} + d_{xz} d_{yx} d_{zy}} \\
Pz_{eq} &= \frac{-g_x d_{yx} d_{zy} - g_y d_{xy} d_{zx} + g_z d_{xy} d_{yx}}{d_{xy} d_{yz} d_{zx} + d_{xz} d_{yx} d_{zy}}
\end{aligned} \tag{1.3}
$$

Discussion The control problem of the underlying system (1.1) is however a difficult problem. In the scope of this report we will set up the system in the framework of a parameter identification optimization (PSO), where we search for the parameters that fit the assumptions of the systems and leads to a 'stable enough' solution. In doing so, we use the particle swarm optimization technique, which is inspired from nature. This algorithm is very powerful for finding the global optimal value of a tactic objective function.

1.3 Parameter identification problem to system (1.1)

For given initial conditions x_0, y_0, z_0 of the three species, we look for the combination of coefficients (or parameters) $c = (g_x, g_y, g_z, d_{xy}, d_{xz}, d_{yx}, d_{yz}, d_{zx}, d_{zy}) \in \mathbb{R}^9$ such that the conditions (1.2) satisfy and the equilibrium points in (1.3) are positive numbers. PSO algorithm is then used to minimize the following integral:

$$f(z) = \int_0^T |z'(t)| dt, \tag{1.4}$$

over the domain of c. This objective function refers to the meaning that the population of shepherds is not dramatically changed over time.

2 Particle Swarm Optimization and its application for system (1.1)

2.1 PSO

[1] PSO is an optimization algorithm which is inspired by the cooperative behavior of various species to look for their needs in the search space.
The update in the current algorithm iteration is guided by the personal experience (particle experience), typically indicated by P_{best}, the overall experience (collective or swarm) denoted by G_{best} and the Present movement (with an inertia component). The experiences are applied by multiplying by two factors c_1 and c_2, and two random numbers generated between $[0, 1]$ meanwhile the present movement is multiplied by an inertia factor w varying between $[w_{min}; w_{max}]$. Inertia factor becomes smaller when the iteration goes larger.
The initial population of size NxD is denoted as $X = [X_1, X_2, ..., X_N]^T$. Each particle $X_i(i = 1, 2, ..., N)$ is given as $Xi = [X_{i1}, X_{i2}, ..., X_{iD}]$. The initial velocity of the population is denoted as $V = [V_1, V_2, ..., V_N]^T$ and the velocity of each particle X_i, $(i = 1, 2, ..., N)$ is given as $V_i = [V_{i1}, V_{i2}, ..., V_{iD}]$, where $j = 1, 2, ..., D$.

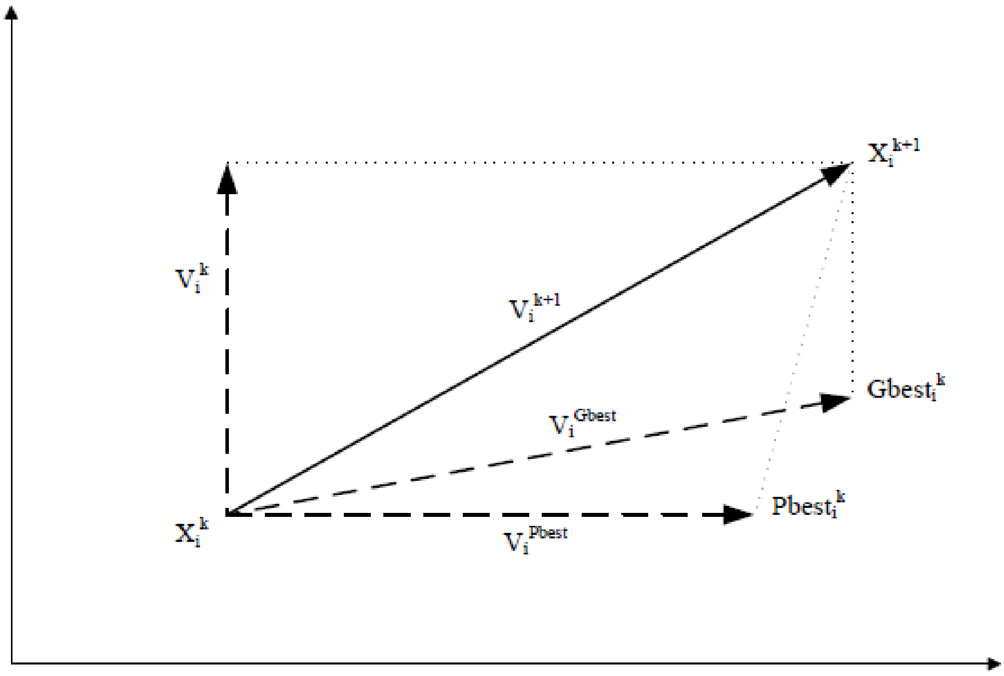

Figure 2.1 PSO search mechanism in multidimensional search space.

2.2 Setup for system (1.1)

PSO applied to system (1.1) is written as follows.

1. Set parameter $w_{min} = 0.4, w_{max} = 0.9, c_1 = 2$ and $c_2 = 2$ for PSO
2. Initialize population of particles having positions X and velocities $V = 0.1$. Population size is $n = 100$. Parameters size is $m = 9$
3. Set iteration k = 1, Maximal iteration is $Maxite = 50$.
4. Calculate fitness of particles $F_{ki} = f(z_{X_i^k})$ and find the index of the best particle b
5. Select $P^k_{best}{}^i = X_i^k$ and $G^k_{best} = X_b^b$
6. Set $\omega = \frac{\omega_{max} - k(\omega_{max} - \omega_{min})}{Maxite}$
7. Update velocity and position of particles

$$V_{i,j}^{k+1} = \omega \times V_{i,j}^k + c_1 \times rand() \times (P_{best_{i,j}^k} - X_{i,j}^k) + c_2 \times rand() \times (G_{best_j^k} - X_{k_{i,j)}}$$

and

$$X_{i,j}^{k+1} = X_{i,j}^k + V_{i,j}^{k+1}, \forall i, j$$

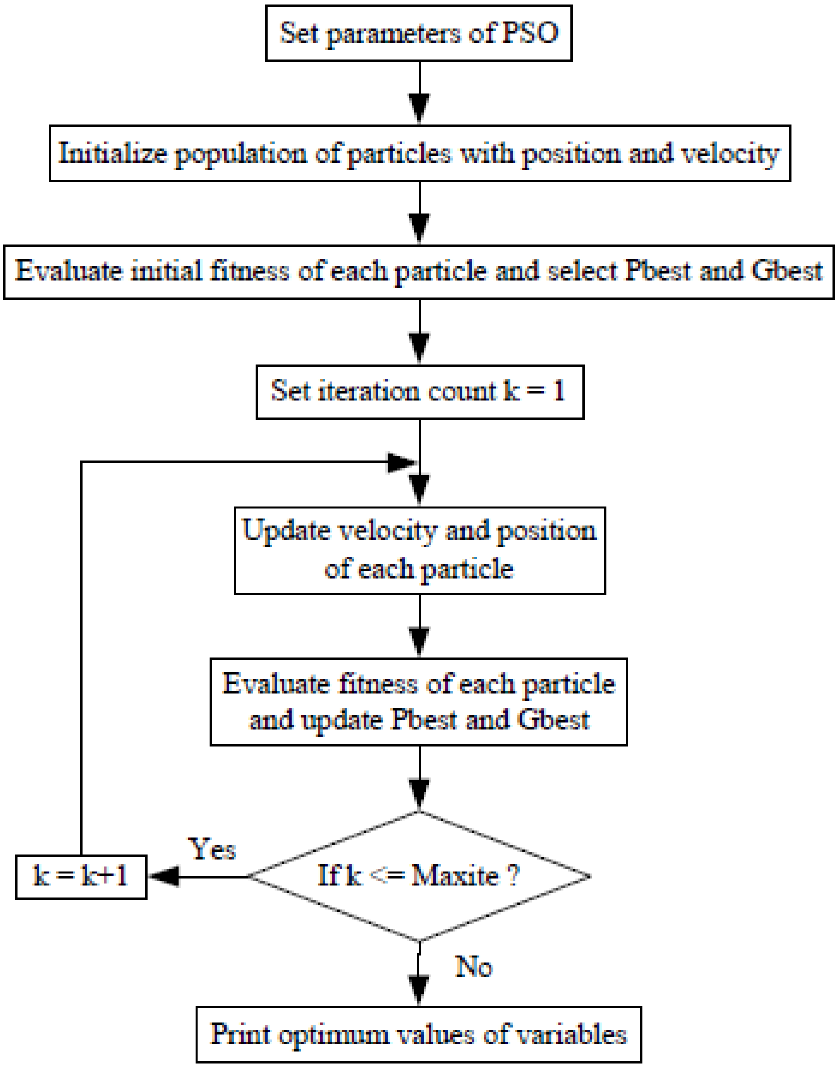

Figure 2.2 Chart of PSO algorithm

8. Evaluate Fitness $F_{k+1,i} = f(X_i^{k+1})$ and find the index of the best particle b1

9. Update $P^k_{best}i$: If $F_{k+1i} < F_{ki}$ then $P^{k+1}_{best}\,i = X_i^{k+1}$ else $P^{k+1}_{best}\,i = P^k_{best}i$

10. Update G_{best} of population If $F^{k+1}_{b1} < F^k_b$ then $G^{k+1}_{best} = P^{k+1}_{best}\,b1$ and set $b = b1$ else $G^{k+1}_{best} = G^k_{best}$

11. If $k < Maxite$ then $k = k + 1$ and go to step 6, else go to step 12

12. Print optimum solution as G^k_{best}

2.3 Results

PSO outputs a local minimum of the objective function. The initial objective function is 6.510^5 with $c = [4.7500 - 3.6200 - 12.7400 - 16.690019.46001.4300 - 5.10004.8000 - 11.3200]$, the local minimum after running PSO is 5.610^5, which reaches at $c_{opt} = [4.0000 - 3.5622 - 12.0000 - 16.949620.00001.4138 - 5.00004.0000 - 11.0000]$; however this combination of coefficients does not lead to a stable system (1.1). In order to deduce stability of the system, we need further research. Following are the graphs of Population of sheeps, wolfs and sherpherds over time.

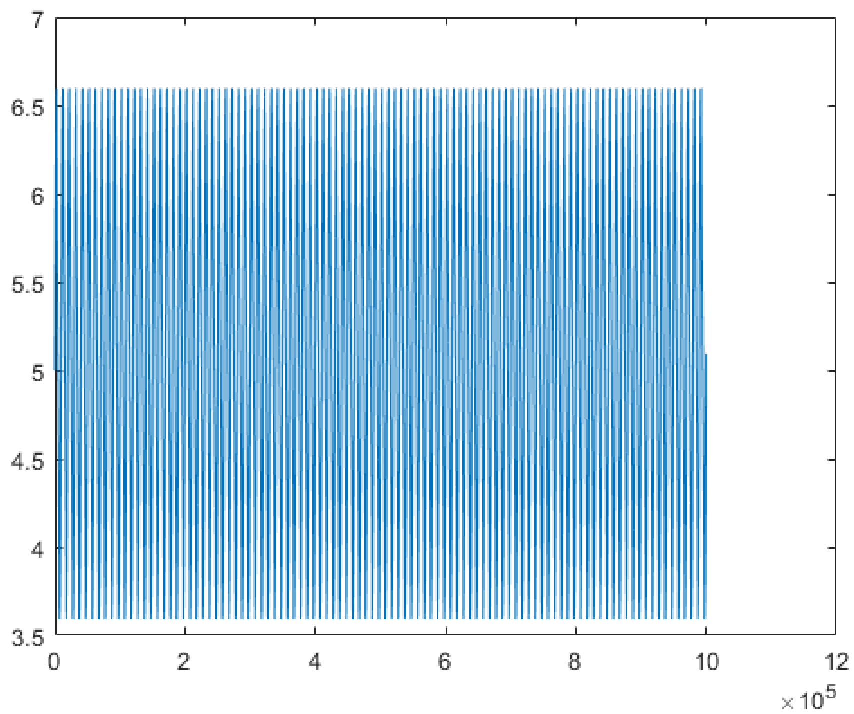

Figure 2.3 Population of sheeps over time

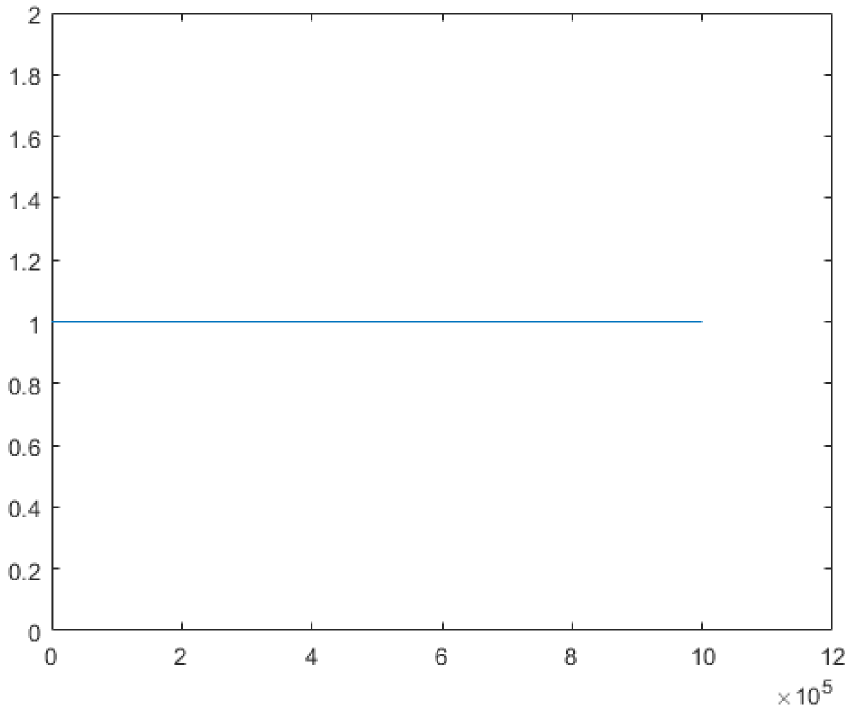

Figure 2.4 Population of wolfs over time

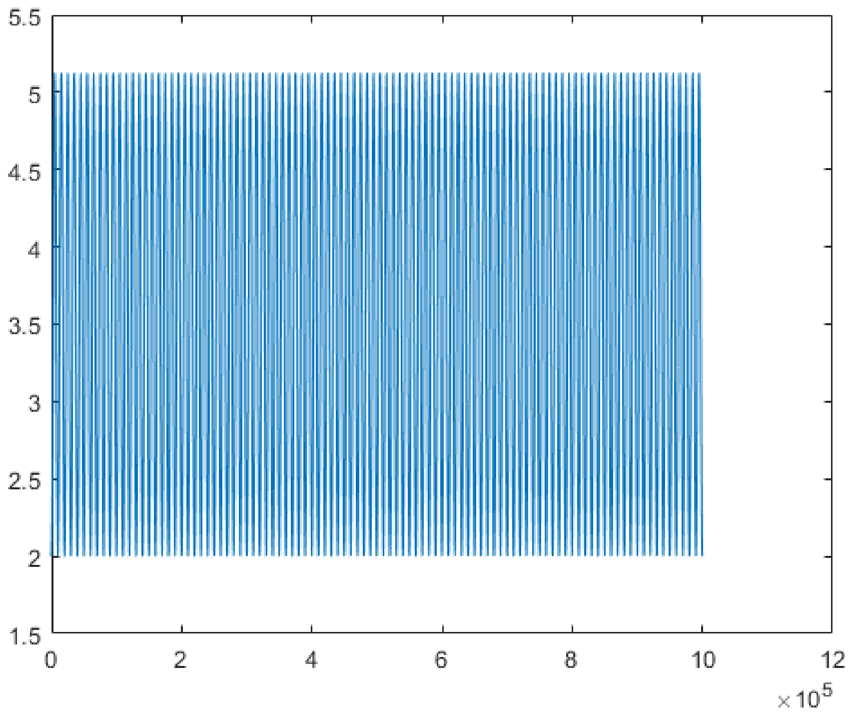

Figure 2.5 Population of sherpherds over time

List of Figures

List of Tables

Bibliography

[1] N. A. Mahamad. 'Particle Swarm Optimization: Algorithm and its Codes in MATLAB'. In: *Project: Application of operation research on solving electrical engineering problems* ().

[2] K. Stephen. 'Model and Control of Lotka-Volterra Three Species Prey - Predator - Top Predator in Specific Case of Sheep - Wolf - Shepherd'. In: *Report on Module Modelling physical systems* (2021).