Study on Audouin's gulls in Delta de l'Ebre using Genetic Algorithms

Author: Joan Pareras Velasco

(Dated: January 7, 2022)

Abstract: Audouin's gulls population at La Banya, in Delta de l'Ebre, have characterized between 2006-2017 a migration tendency. The aim of this article is to characterize this behaviour using Genetic Algorithms fitting our proposed solution with the optimal parameters. Our proposal is that this tendency is caused by social copying.

I. INTRODUCTION

At La Banya, gulls build their nest in sub-colonies. There are two types of gulls, Ardouin's gulls and their main competitor yellow-legged gulls. Censuses are performed yearly to control the size of each sub-colony.

The obtained data is shown in the following table:

year	pop.	year	pop.	year	pop.	year	pop.
1981	36	**1990**	4300	**1999**	10189	**2008**	13031
1982	200	**1991**	3950	**2000**	10537	**2009**	9762
1983	546	**1992**	6174	**2001**	11666	**2010**	11271
1984	1200	**1993**	9373	**2002**	10122	**2011**	8688
1985	1200	**1994**	10143	**2003**	10355	**2012**	7571
1986	2200	**1995**	10327	**2004**	9168	**2013**	6983
1987	1850	**1996**	11328	**2005**	13988	**2014**	4778
1988	2861	**1997**	11725	**2006**	15329	**2015**	2067
1989	4266	**1998**	11691	**2007**	14177	**2016**	1586
						2017	793

FIG. 1: This table shows the yearly performed censuses of Arduin's gulls at La Banya.

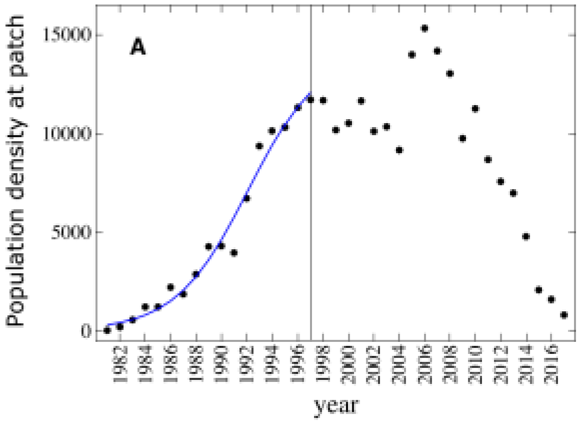

FIG. 2: This figure shows graphically the data with the First Epoch model adjusted.

From this data two main epochs can be differentiated. The First Epoch which is characterized by a logistic growth due to the absence of predators and the the fact that the population did not exhaust the food carrying capacity. And the period between 2006–2017 known as Second Epoch, which is characterised by a regular behaviour with migration.

As shown in the FIG 2, this first epoch has already been characterized. However, the Second Epoch have to be characterized also.

Regarding the Second Epoch, there is no evidence of a decrease of food, or an increase in competitors or predators that explain the decrease of population. Then this decline must depend on other factors.

Our hypothesis is that the presence of predators causes a dispersal by social copying. That is, the smaller the population the higher the dispersion.

II. MATHEMATICAL MODEL WITH DISPERSAL BY SOCIAL COPYING

Our model has already been adapted on the First Epoch as shown in FIG. 2, and has to be now implemented to the Second Epoch. The model will consider the full collapse of population during this period, as in 2017 the predators were removed. The model reads as

$$\frac{dx}{dt} = \alpha x - \gamma x^2/K - \lambda \Psi(x,\mu,\sigma,\delta) \tag{1}$$

Where the first term refers to the immigration, growth and death, the second term to the nonlinear competition and finally the dispersal by social copying. The competition term can be rewritten as βx^2 where $\beta = \sigma/K$. The value of $\beta := 0.000024382635446$ has been computed for the First Epoch, and therefore, it will be the same for the Second Epoch. The first two terms represent a logistic growth, whereas the nonlinear dispersal function by social copying is given by:

$$\Psi(x,\mu,\sigma,\delta) := \begin{cases} \frac{1-\mathcal{E}_{dir}(x,\mu,\sigma,\delta)}{1-\mathcal{E}_{dir}(0,\mu,\sigma,\delta)} & \text{if } 0 \leq x \leq \delta \\ \frac{1-\mathcal{E}(x,\sigma,\delta)}{1-\mathcal{E}_{dir}(0,\mu,\sigma,\delta)} & \text{if } x \geq \delta \end{cases} \tag{2}$$

$$\mathcal{E}_{dir}(x,\mu,\sigma,\delta) := \left(\mu \frac{\Theta + \sigma\delta}{2\Theta + \sigma\delta}\left(1 - \frac{x}{\delta}\right) + \frac{x}{\delta}\right)\mathcal{E}(x,\sigma,\delta) \tag{3}$$

where,

$$\mathcal{E}(x,\sigma,\delta) := \frac{\sigma(x-\delta)}{\Theta + \sigma|x-\delta|} \tag{4}$$

is an Elliot sigmoid Θ-scaled, σ-strengthened, and δ-displaced. In this case, we have fixed $\Theta := 1000$.

Our algorithm will compute the value of the different parameters to achieve the best fitted function Ψ. This parameters are the following:

- $x(0)$ the value of x at 2006. This value is also going to be fixed at $x(0) = 15329$ as it is the census value at that time.
- ϕ which must be $\phi \leq \alpha = 0.3489494085776018$, is the neat population growth rate.
- λ is the non-linear dispersal rate.
- μ determines the sign of $\frac{d}{dx}\Psi(x, \mu, \sigma, \delta)|_{x=0}$.
- σ determines the slopes of the sigmoids.
- δ point of the change of concavity of $\Psi(x, \mu, \sigma, \delta)$.

III. GENETIC ALGORITHM'S

A Genetic algorithm is a metaheuristic inspired by the natural selection process as explained in [1] and [2]. This algorithms are commonly used to generate high-quality solutions to optimization and search problems. The algorithm uses biologically inspired operators such as selection, crossover and mutation.

For this problem, the algorithm has been adapted to find the optimal parameters so our solution fit the best our data.

The structure of genetic algorithms is as follows:

```
Genetic Algorithm
  Initialization of population: N random genotypes
  Compute fitness of each individual
  while not convergence and $iter < Max_{iter}$ then
    Selection fittest → New population
    Crossover of parents with a high probability
    Mutation with low probability of each individual
    Compute fitness of each individual
    if Ending condition True then
      end while
  end while
```

Notice, the algorithm has four different parts named: selection, crossover, mutation and ending condition. Each of these parts can be done with different methods. The methods used in our algorithm will be presented in the following subsections.

A. Genotype

The word genotype is the name given to identify each individual of the population. Is the analogue to a genome in biology and characterize each individual as an unique combination of chromosomes. In our problem, each chromosome refers to each parameter. As iterations go by, the population will become fitter. This means that the algorithm will give the best possible solution to our problem found.

B. Selection

The genetic algorithms, as in natural selection processes, are characterized by evolving towards the best adapted. While coding, this has to be reflected in the selection of which individuals will evolve. If in each iteration the individuals that we select to survive and reproduce are the fittest ones, the population will converge to the best solution found.

In this case, the selection method utilized, has been the tournament selection method. This method works as follows:

```
Tournament Selection
  initialize empty nextgen with size N
  for N individual do
    for tournament size times do
      select random newindividual from population
      compute fitness of newindividual
      if fitness[newindividual] < best fitness then
        best fitness = fitness[newindividual]
        best individual = newindividual
      end if
    end for
    sort by fitness
    start counter
    do while random value < $prob(1 - prob)^{counter}$
      nextgen[individual] = newindividual[counter]
      counter++
    end do while
  end for
```

Notice that the new population will be a selection of individuals selected randomly and with a certain probability of being the best of each tournament. If *prob* value is high, most of the times the elected value will be the fittest of the k individuals that take part on the tournament. This probability is important to be different of 1 to be able to explore other regions of solutions, that may have fitter solutions.

C. Crossover

Crossovers is similar to human reproduction. A combination of gens coming from each individual parent. There are different methods that permit this procedure.

In the case of the genetic algorithms, the parameters to optimize are represented as unsigned integers expressed in binary. Taking into account the searching range and the precision accepted for each parameter. Then, the crossover of chromosomes can be performed as a serial of binary operations.

In our algorithm, three possible options can be chosen: one point crossover, two point crossover or uniform crossover.

The one point crossover breaks each parent chromosome in two parts of random lengths, then the first part of the first parent is attached with the second part of the second parent, and vice-versa. Therefore, two new chromosomes of the same length as the previous ones are created.

The two point crossovers follows the same procedure as the one point crossover method, but breaking the chromosomes in three parts.

Finally, in the uniform crossover, each bit is chosen from either parent with equal probability.

Crossovers in genetic algorithms are performed with a certain probability, typically between 0.6 and 1., otherwise the next generation individual is the same as its parent. This values is chosen taking into account our approach to solve the problem.

D. Mutation

Mutations in genetic algorithms consist in changing randomly a bit of each chromosome with a certain probability. This way other regions of the possible chromosomes can be explored. Usually this mutation ratio is lower than 0.1.

In our code, the mutation is done using the Bit Flip Mutation.

E. Ending condition

Typically, the ending condition that needs to be fulfilled is that the 90% of the population lays around the same chromosomes. However, due to the complexity of our code, another condition has been imposed. If after a certain number of iterations, the algorithm has not found a better fit, then the algorithm stops. This number can be changed taking into account your approach and computation resources.

IV. RESULTS

In this section, the results obtained with our algorithm will be presented.

A. Convergence study

Our algorithm is designed to find the best fit will exploring other possible parameters with better fitness. This error is computed as the quadratic error between the real data and the predicted EDO solutions using Runge-Kutta Ferberg 7-8 method. Regardless, a remaining question has to be answered, how quick is our algorithm to converge? To answer this question the best fit at each iteration has been plotted in the FIG. 3.

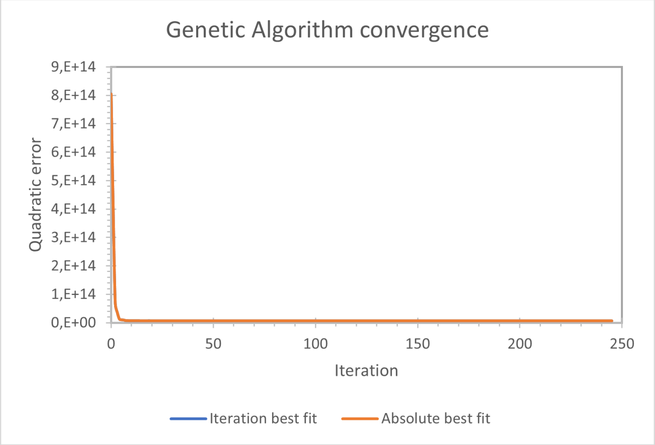

FIG. 3: This graphic shows the convergence evolution at each iteration step.

Notice the algorithm rapidly converges to fitter values.

Another possible consideration is which is the effect of applying the different crossover methods. For both, one point crossover and two point crossover the convergence is almost equal. In contrast, the uniform crossover is much more exploratory, but is more inefficient to find better solutions than the other two methods.

The size of the system also has an effect on the convergence to the optimal solution. For higher populations, the number of explored values in each iteration increases giving a higher probability to find a better solution at each step. However, as higher is the system, more computationally expensive the algorithm becomes.

B. Best solution found

As previously explained, the algorithm will probably find a solution which is not the best option. However the value obtained is good enough to explain the behaviour of the system. Also mention that in all executions the found values are within the same value range.

After doing different executions the best fitted solution that has been found is:

Parameters	Optimal Value
Phi	0.22495812
Lambda	1828.3530
Mu	1.034989
Sigma	2.77
Delta	10385

TABLE I: This table shows the best found solution.

This result has been obtained considering:

- Population of 20000 individuals.
- Tournament selection of 50 individuals with probability of selection of 0.8.
- One point crossover with probability of 0.95.
- Bit flip mutation with a probability of 0.08.
- Ending condition of 200 iterations being the fittest result.

The predicted data has been represented in FIG 4 in comparison with the real data.

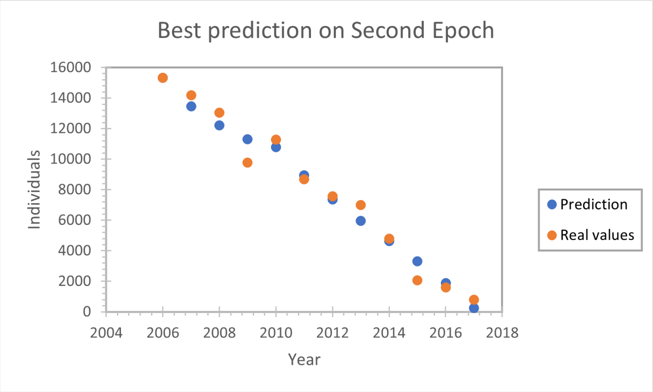

FIG. 4: This graphic shows the real number of gulls vs the predicted value with our algorithm.

Notice, that the model fits the data satisfactorily except from some values that present some significant anomaly.

V. POSSIBLE IMPROVEMENTS

The algorithm uses the tournament selection method to select the fittest parents, but it is not the only possible method. A possible improvement, will be to add other selection methods as elitism, rank selection... and see if using them we find a fitter solution. In any case, the tournament selection is a great choice as it is exploitatory will also being exploratory.

Another possible improvement will be to study which are the best parameters for crossover and mutation, although the ones used are in a typically good range. With different combinations of these parameters the algorithm can be more exploratory or more epxloitatory.

VI. INSTRUCTIONS FOR COMPILING

To compile the algorithm the only needed step is to compile and execute the main.c algorithm. If needed in the functions.h file the parameters used in the code can be changed. There is a README file explaining each function and definition used in the code.

The delivered file, contains:

- Genetic Algorithm
  - data
  - src
    - main.c
    - functions.h
    - RKF78.h
    - RKF78.c
    - random.h
    - Crossovermutation.h
  - Genetic Algorithm.pdf
  - README

VII. CONCLUSIONS

In conclusion, our algorithm approach permits finding good solutions and converges in less than 100 iterations, when considering a sufficiently large population. The solution found is never the same. However, the quadratic error is always in the same range and the rank of the parameters is always the same. For further improvements, a study in the characterization of the algorithm must be done.

[1] M. Melanie, *An Introduction to Genetic Algorithms* (A Bradford Book The MIT Press, 1999), 5th ed., ISBN ISBN 0262133164 (HB), 0262631857 (PB), available for free at https://books.google.es/books?hl=eslr=id=0eznlz0TF-ICoi=fndpg=PP9dq=cite+an+introduction+to+genetic+algorithmsots=simB6–dNdsig=lGV6iCfjxXYTvNh9h4qDDq3Qd0ov=onepageqf=false.

[2] S. Luke, *Essentials of Metaheuristics* (Lulu, 2013), 2nd ed., available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.