For office use only	Team Control Number	For office use only
T1 _____	**1911850**	F1 _____
T2 _____		F2 _____
T3 _____	Problem Chosen	F3 _____
T4 _____	**A**	F4 _____

2019
MCM/ICM
Summary Sheet

When The Dragon Becomes A Reality

Abstract

The dragon will be a wonderful memory for your son or daughter, and our model will make dragon training a reality, if the dragon is real.

We start with the growth of dragons, assuming that the indicators that can truly reflect the growth of dragons are related to functions, weight, diet, changes or other characteristics related to animals. This is a multivariate nonlinear objective programming model. After data transformation, the model is established for solution. With the growth of the dragon, the total amount of environmental resources is constantly changing, which means that the total growth will slow down, and the retarded growth model will follow.We got a lot of valuable data, such as the ecological demand of the dragon: 13.54 sheep/day, 808.26kg of vegetables/day. The population density of sheep was $0.548/km^2$.

The ecological impact of dragons is a problem within the planning. We made assumptions about the energy demand based on the data of raptors. The region where the dragon lives can provide different levels of help to the dragon. Collect regional information, and use the method of dimensionality reduction to classify different regions. We have also expanded from one dragon to many, taking into account the impact of different climatic regions. The living space a dragon needs is $824km^2/day$. Moreover, the interaction between dragons is studied innovatively, considering the competition between dragons and the attraction between dragons of the opposite sex.

The migration of the dragon is an important part of the survival of the species. Multivariate analysis is used to determine the ecological factors between different environments. The impact of the change can be determined by establishing a corresponding analysis between the migration of the dragon and the resources needed for its growth.

In addition, we analyze the performance of our model under various conditions, and modify our model to accommodate to them. We also mixed C++ and Python, using TDD architecture, greatly improve the program extensibility and readability. To sum up, our model is a feasible and reasonable model which can accommodate to various situations.

Contents

1 Introduction

1.1 Background Situation

Dragons are massive, flying reptiles that can breathe fire onto their enemies and cook their food with the same flame. First of all, we should be clear that the dragon we're talking about is in the context of the novels *A Song of Ice and Fire*. They are rumored to have a strong connection to magic, which seems to be proven true when magic begins to return to the world after the birth of the first three in over two hundred years. Dragons possess awesome and devastating power, capable of laying waste to armies and burning entire cities to ashes. Men who were able to tame and ride dragons as beasts of war used them to burn their enemies and forge vast empires across the continents of Essos and Westeros.

Dragons have long serpentine bodies, with proportionately long necks and tails. Their bodies have four limbs: two short back legs and two large wings as forelimbs, a body-plan similar to a bat. In later generations, after the dragons went extinct, physical descriptions of dragons became so confused in memory that artwork sometimes depicted them as having six limbs — two wings growing out of their backs in addition to four legs — but this is inaccurate. The teeth and claws of adult dragons are as long and sharp as swords.

Dragons are covered in scales, as well as spines that run down their backs from head to tail. Particularly large ridges of horns frame the edges of their faces, running along the back of the skull and along the jawline, which grow bigger as they mature. Adult dragons possess two sets of frills that run along the backs of their necks and spine, two along the sides of their necks and another two centered closer to the backbone, for a total of four frills. These are formed from webbing that grows between longer spines. When dragons are agitated (or simply excited), they raise and flare these frills — similar to how a furry animal like a cat will raise the hackles on its back when agitated (or a feathered animal such as a goose will puff up its feathers), in an attempt to appear bigger so as to intimidate its enemies.

Dragons are also shown to have a variety of calls, from shrieking roars to low growls or hisses. They can even squeal.

Dragons are obligate carnivores, with diets consisting entirely of meat. Dragons need to roast their prey with their fire-breath before consuming it — the only animals apart from humans who prefer cooked meat. Dragons can eat almost any kind of meat, anything from sheep to fish. Historical dragons ridden as beasts of war were known to eat fallen horses and even men on the battlefield. Fully grown dragons could swallow a live horse whole[1].

1.2 Problem Restatement

Dragons are animals that do not exist in the real world, but interesting mathematicians can make the realm of fictional dragons real by the mathematical modeling. We first have to go through the analysis of dragon's characteristics, behavior, habits, diet and interaction with the environment and other basic issues. At the same time, the dragon's ability to fly, breathing fire principle, the ability to resist trauma are also worth studying. Knowing the basic situation of dragons, we can answer the following questions well. What is the environment for the dragons to grow? How to calculate the calorie intake and consumption of a dragon? How much living area should be provided and what community support should be provided in the face of multiple dragons living together? At the same time, the influence of climate factors on research results should not be ignored. We also have to consider the possibility of migration. In addition, we can expand and consider more problems that dragons may face in today's life, and use modeling work to explore the ecological benefits of dragon breeding.

2 Model Assumptions

2.1 Dragon Characteristics

(1) Dragons are omnivorous animals, but mainly carnivorous.

(2) We think the dragon is at the fifth level of the energy level and is the most advanced consumer.

(3) Dragons like to roast their food in dragon flame before eating.

2.2 Dragon Living Environment

(1) Without considering the actual climate conditions, we first built the model of living environment for the dragon according to the environment of the nature reserve.

(2) Dragons have no tendency to live in groups, and although there are occasional groups of dragons, most of the time they live alone.Dragons are also willing to fight another dragon if needed.

2.3 Dragon Abilities

(1) Dragon scales are one of the most important protective measures for dragons.We divide trauma into two categories, physical and magical. Since the situation is set in today's environment, we only consider the physical damage that dragons might encounter in daily life.

2.4 Dragon Growth

(1) Dragons grow all their lives. We believe that the size of dragons is limited by their growing environment. Therefore, the dragons growing in dragon dens do not grow to the size of their wild ancestors. But in general, older dragons are bigger.

(2) As dragons get older, the dragon scales get thicker and thicker.

2.5 Symbol Table

Please see the table 1.

3 Basic Model Design and Analysis

3.1 Dragon's Weight Model

Definition 1. *Weight The body weight of a dragon. Weight depends on its food condition and quantity, which affects its ability.*

We think that weight is an important attribute of the dragon, and the dragon's ability is related to the weight of the dragon. Regardless of environmental constraints, the weight of a dragon increases with age. According to the background data, the weight of the dragon was 10kg at birth and 30-40kg a year later, and it was assumed that the adult weight was about 5–10t. Substitute the data to verify which model's curve law is more consistent with the growth curve of the dragon.

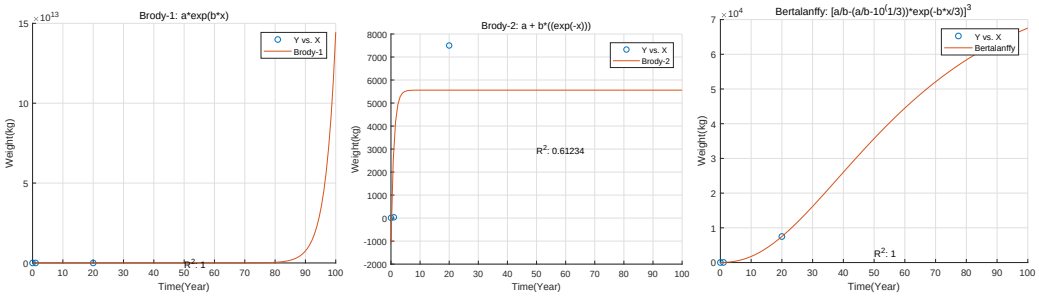
Note:In this model, A is the mature weight,W_0 is the birth weight, and the rest letters represent parameters.However, the same parameter may represent different meanings in different models.

After data substitution, figure 1(f) is the most consistent with the actual situation.

3.2 Dragon's Energy Model

Definition 2. *Energy*

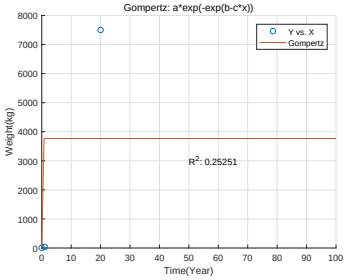
Methane is a high-energy substance produced by the fermentation of carbohydrates in the feed in the rumen. Ruminants need to repeatedly absorb nutrients, and there are small peptides in the rumen, which promote the generation of gastric microorganisms, such as methanogens, which will decompose carbohydrates and generate methane.



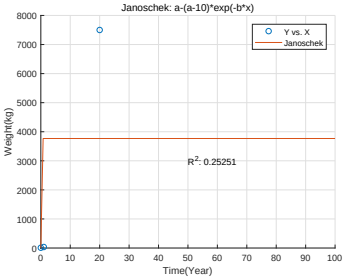
(a) Brody-1

(b) Brody-2

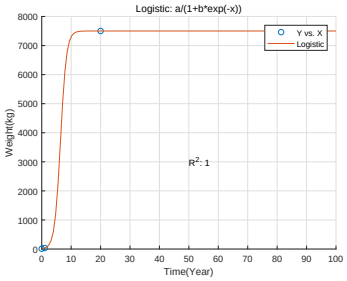
(c) Bertalanffy



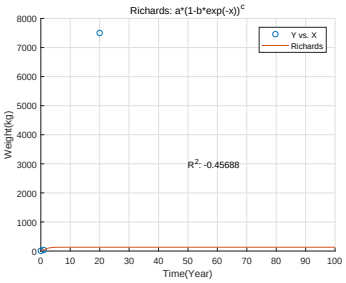
(d) Gompertz



(e) Janoschek



(f) Logistic



(g) Richards

Figure 1: Fitting the weight growth curve of the dragon

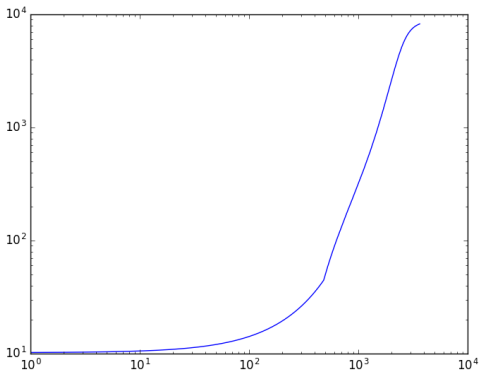


Figure 2: Weight in the first 10 years , plotted on a logarithmic axis

Symbol	Explanation
Y_m	The amount of methane produced per 100MJ of feed(MJ).
D	Apparent digestibility of feed (coarse feed is adopted for ruminants such as cattle and sheep, 60 is often taken).
L	Intake level, the ratio of energy intake to maintenance energy
M	The quality of methane(kg/a/per).
A	Activity coefficient(Since the adult cow is 1.2, and the cattle is 1.1, 1.5 is taken for the dragon).
NE_f	Net energy required for flight(MJ/day).
NE_m	Net energy required for maintenance(MJ/day).
NE_g	Net energy required for growth(MJ/day).
NE_a	Net energy required for activity(MJ/day).
NE_h	Net energy required for fire breathing(MJ/day).
GE	Animals feed on total energy(MJ/day).
SE	Dragon daily defecation energy(MJ/day).
W	The weight of the dragon(kg).
W_g	The daily gain of the dragon(kg).
T_f	The average daily flight time of a dragon(h).
T_s	The moment when dragon energy reaches critical value(h).
E	The mental state coefficient of the dragon.
V	The speed at which a dragon flies in its daily activities(km/h).
H	Daily activity time(h).
X_0	The magnitude of vegetables weight intook(kg).
Y_0	The magnitude of mountain goats eaten(one).
ρ	Population density of dragon food.

Table 1: Symbol Table

The proportion of energy the animal consumes converted to methane: [3]

$$Y_m = 1.30 + 0.112D + L(2.37 - 0.050D) \tag{1}$$

Daily methane emission from animals:

Consumption of energy: assuming that dragons are warm-blooded animals, the energy of maintenance, growth and activity can all refer to domestic cattle. The formula coefficient can be modified according to the actual data.[4].

- Net energy required for maintenance(NE_m)
  - Adult dragon: $0.299w^{0.75} \times A$.
  - Growing dragon: $0.614w^{0.67} \times A$.
- Net energy required for growth(NE_g): $WG(1.5 + 0.045W)/(1 - 0.030WG)$. Ignore the adults.

Model	Formula	Time to the inflection point	The weight at the inflection point
Brody	$W = W_0e^{kt}, W = a - be^{-kt}$	$\cdots$	$\cdots$
Logistic	$W = a/\left(1 + be^{-ct}\right)$	$\ln b/c$	$a/2$
Gompertz	$W = a\exp\left[-e^{b-ct}\right]$	b/c	a/e
Richards	$W = a\left(1 - be^{-kt}\right)^m$	$\ln(bm)/k$	$a\left[(m-1)/m\right]^m$
Janoschek	$W = a - (a - W_0)\, e^{-ktp}$	$\left[(p-1)/(pk)\right]^{1/p}$	$a - (a - W_0)e^{\frac{1-p}{p}}$
Bertalanffy	$\left[a/b - \left(a/b - W_0^{1/3}\right) e^{-\frac{bt}{3}}\right]^3$	$2/b\ln\left[3 - 3b/aW_0^{1/3}\right]$	$8a/27$

W_0 is the initial weight, a is the weight of mature dragon. b,c,m,p are parameters

Table 2: Characteristics of different growth curves[2]

- Net energy required for activity(NE_a):

$$NE_a = \int_0^{T_f} 6.468 \times \frac{A}{E}\,\mathrm{d}t.$$

- Net energy required for breathing fire(NE_h): $0.7M \times 50.07$, take the calorific value of methane is $50.07MJ/kg$,take its combustion rate is 70%.

- Daily energy intake(Suppose the energy conversion efficiency is 55%).

$$GE = \frac{NE_m + NE_g + NE_a + NE_f + NE_h}{0.55 \times D\%} + SE.$$

3.3 Dragon's Mental State Model

Definition 3. *Mental State*[1]

Denote as E

(1) The efficiency of dragon is a number changing between $(0, 1)$, which is related to its daily activities, and the independent variable is time.

(2) The efficiency of dragons affects the energy consumption of their daily activities. Specifically, the higher the mental state, the lower the energy consumption.

(3) The efficiency decay rate of the dragon is related to its age, that is, the older the dragon is, the slower its mental state consumption will be (reflected by the decrease of Q value).

[1]Can be viewed as efficiency

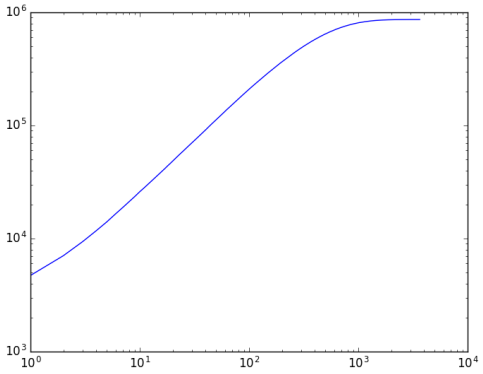


Figure 3: Ground energy in the first year to the tenth year

(4) The dragon's energy should be independent of each other every day, and the energy should be restored to the threshold E_0 on the second day after sleeping every night (represented by the random number generated between $(0.9, 1)$, indicating that the dragon's sleep is inconsistent every day, and the E_0 is high if the dragon sleeps well).

(5) The irregular work and rest of the dragon is reflected in the fact that when E decreases to a certain critical value (denoted as E_f), the dragon will start to rest, and when $E < E_f$, the dragon will stop moving.

E_f range reference $(0.1, 0.15)$, we can get the figure 4 plotted.

$$E = E_0 e^{-kt} \tag{2}$$

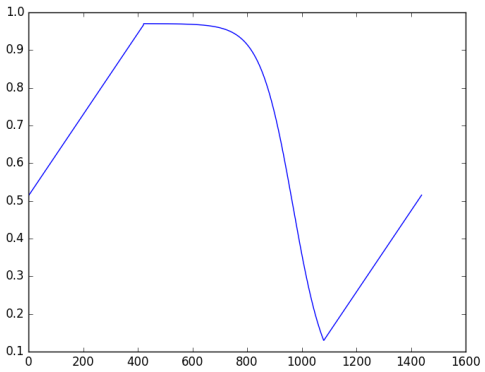
and the value Q is

$$Q = \frac{A}{1 + Be^{CT}} \tag{3}$$

- A, B and C are constant in the equation(3), which should be determined according to the data
- T: the age of the dragon
- t: the time when the dragon is active after getting up, $t = 2$ represents the second hour after getting up
- T_f: the moment when dragon mental state reaches the critical value

3.4 Fly Model

The flying process of the dragon is regarded as uniform motion, regardless of the influence brought by acceleration, temperature change, flying height and efficiency.



Gradual consumption in daytime, linear recovery at night

Figure 4: Mental state – efficiency

3.5 Breath Fire Model

1. The fire-breathing principle assumes that the rumen in the dragon body can decompose the cellulose in food to produce methane, which is stored in a certain organ in the body. When it is necessary to breathe fire, the methane will be transported to the lower jaw, and the flame will be ejected through ignition(To do this, it is also necessary to assume that the dragon's food contains green plants, otherwise cellulose cannot be obtained.).

2. The fuel consumption of the dragon does not consider the effect of efficiency.

The proportion of energy the animal consumes converted to methane: [3]

$$Y_m = 1.30 + 0.112D + L(2.37 - 0.050D)$$

Feeding level calculation

$$L = A\frac{NE_m + NE_g + NE_a + NE_f}{NE_m}$$

Daily consumption of methane

$$M = NE_m\frac{Y_m}{100} \times 0.3049$$

3.6 Resist Trauma Model

As mentioned in the hypothesis, dragon scales are an important defense for dragons.Besides, the claws and wings of dragons are not only weapons of attack, but also means of defense.The dragon is able to withstand great trauma, and the dragon's ability to heal itself is also very strong.So unless it's a devastating injury, the dragon is essentially unaffected.

3.7 Climatic Condition Simulation Model

Note: Since the dragon's hard shell is not easy to be affected by external temperature, the direct impact of temperature on the life of the dragon is not discussed in this model (there is still an indirect impact).

Definition 4. *Arid Region Features: lack of water day and night, temperature difference, more sand, strong sunshine.*

Impact on dragons:

1. Windblown sand and sunshine hinder flight: appropriately increase the flight energy consumption coefficient and appropriately reduce the average flight speed V.
2. Because of the hot environment and lack of water, the appetite and digestion level of the dragon are reduced: both the food energy GE and the digestibility are reduced.
3. As plants are scarce, dragons cannot get enough cellulose from their ingestion: proper reduction of the conversion rate of methane in Y_m formula will indirectly reduce dragons' fuel consumption[2].

Definition 5. *Warm Temperate Region Features: comfortable climate, abundant resources.*

Impact on dragons:

(1) Due to the appropriate environmental conditions, the dragon's energy decline rate will slow down: the K value[3] will be appropriately reduced.

(2) Adequate food sources: appropriately increase the dragon's food intake GE and fecal mass SE.

Definition 6. *Arctic Region Features: extremely low temperature, mostly Marine glaciers.*

Impact on dragons:

1 Due to the extremely cold temperature and the decrease of food sources, it is considered to add the hibernation mechanism for dragons in this region (sleeping most of the time, the energy consumption for flight, activity and fire breathing is greatly reduced, but the energy consumption for growth and maintenance remains the same);Specifically, the dragon efficiency threshold is raised, and E_f is modified to about 0.8 (that is, the daily rest time is greatly advanced, and the activity time is greatly reduced) : it is not necessary to be 0.8, and the ideal E_f value should make the daily activity time about two to three hours.

[2]The energy expended by breathing fire.
[3]Maximum environmental capacity.

2 Due to hibernation, the daily flight time is greatly reduced, and the T_f is adjusted to about 10min-30min.

3 Because the temperature is too low, it has an inhibitory effect on the fire spraying effect of the dragon, and the emitted flame is weakened (the coefficient is reduced to 0.2-0.4 in the formula of the fuel consumption, that is, the combustion rate is 20% to 40%).

4 GE energy intake should be reduced by the same proportion, that is, daily food intake should be reduced during hibernation.

3.8 Population and Community Model

Next, we will discuss the impact of the addition of dragons on other populations in the environment. Note:

(1) Since it is assumed that the dragon lives in a natural reserve (Yellowstone national park), its main food source should be cattle, sheep and other herbivores.

(2) Since it is assumed that the fire breathing principle of the dragon is that the body organs convert the ingested cellulose into methane, it is assumed that the dragon also feeds on grass in this setting.

It is assumed that dragons mainly feed on cattle and sheep, and now we can explore its impact on the food chain (only the food chain including cattle and sheep is considered).

It is assumed that in the original environment:

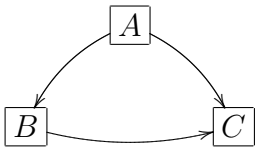
producer A (green plant)

consumer herbivore B (cattle and sheep)

now put dragon C is added to the food chain

The competitive relationship between cattle and sheep is not considered here, and it is regarded as the same category.

.



Before the dragon joined the food chain, A and B formed the standard predation model

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (a - by)x \tag{4}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = (nx - m)y \tag{5}$$

By studying the orbits of the equations (4) and (5), we can know the quantity change rule of A and B.

The equation set has two singularities $(0,0)$, $(m/n, a/b)$, and the value of the Jacobi matrix of the vector field of the equation set (4) and (5) at the singularities $(0,0)$ is

$$J = \begin{bmatrix} a-by & -bx \\ ny & nx-m \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & -m \end{bmatrix} \tag{6}$$

The two eigenvalues of J are $a > 0$ and $-m < 0$, so the singularity $(0,0)$ is the saddle point and unstable.

The value of the Jacobi matrix of the vector field of equations (4) and (5) at the singularity $(m/n, a/b)$ is

$$J = \begin{bmatrix} a-by & -bx \\ ny & nx-m \end{bmatrix} = \begin{bmatrix} 0 & -bm/n \\ na/b & 0 \end{bmatrix} \tag{7}$$

The two eigenvalues of J are imaginary $\pm i\sqrt{ma}$, so the singularity $(m/n, a/b)$ is the focal singularity. The equation is reduced to first-order differential equations

$$(nx-m)y\,\mathrm{d}x = (a-by)x\,\mathrm{d}y \tag{8}$$

Equation (8) is a variable separation equation with zero solutions($x = 0$, $y = 0$) and two semi-linear orbits

$$\begin{cases} x = 0, y > 0 & \text{Rail line:} y = y_0 e^{-mt} \\ y = 0, x > 0 & \text{Rail line:} x = x_0 e^{at} \end{cases}$$

Separable variables

$$\frac{(nx-m)\mathrm{d}x}{x} = \frac{(a-by)\mathrm{d}y}{y} \tag{9}$$

From the initial value $(x_0, y_0)(\neq (0,0))$ to (x, y), the definite integral of equation (9) is made to obtain the integral curve passing through (x_0, y_0)

$$-m\ln\left(\frac{x}{x_0}\right) + n(x-x_0) = a\ln\left(\frac{y}{y_0}\right) - b(y-y_0) \tag{10}$$

Take the exponent of the equation (10)

$$y^a e^{-by} x^m e^{-nx} = K \tag{11}$$

Where K is a constant

$$K = y_0^a e^{-by_0} x_0^m e^{-nx_0} \tag{12}$$

Note that

$$f(x) = y^a e^{-by} \tag{13}$$

$$g(x) = x^m e^{-nx} \tag{14}$$

The differential method tells us that $f(x)$ is a unimodal function, which gets the maximum at the focal ordinate $y = a/b$, and gets the minimum 0 for $y = 0$

and $y = +\infty$. $f(y)$ increases from 0 strictly monotone to the maximum on the interval $[0, a/b]$, and decreases strictly monotonically and approaches 0 on the infinite interval $y > a/b$.

In the same way, $g(x)$ is a unimodal function, which gets the maximum at the focal abscissa $x = m/n$, and gets the minimum 0 for $x = 0$ and $x = +\infty$.$g(y)$ increases from 0 strictly monotone to the maximum on the interval $[0, m/n]$, and decreases strictly monotonically and approaches 0 on the infinite interval $x > m/n$.

So the K in the equation (12) has to satisfy the following inequality

$$0 \leq K \leq \frac{a^a m^m e^{-a-m}}{b^a n^m} =: K_0. \tag{15}$$

Through the above fact is easy to know when the equation (11) in the K value $(0, K_0)$, the corresponding orbit type is surrounded by a focal point singularity $(m/n, a/b)$ closed track. Therefore, the equations of singular point $(m/n, a/b)$ is the center. In the first quadrant center filled with surrounded around the center of closed track. This shows that when the initial values x_0 and y_0 are greater than 0, a and b are not extinct, and their number is cyclical change. See below: After

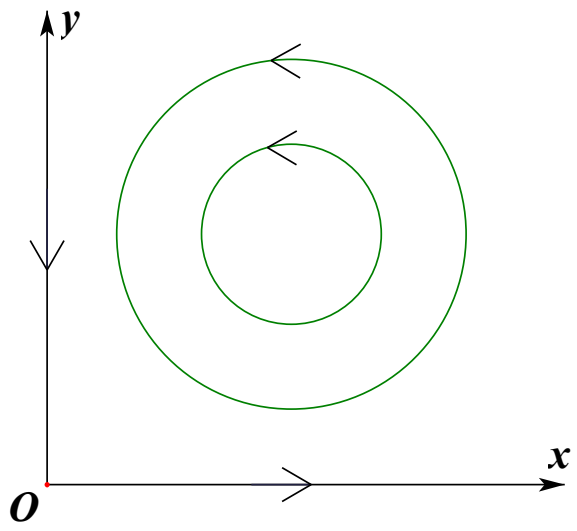


Figure 5: Periodic change

dragon C joins the food chain, the model becomes (in this model, the number of dragons is constant to three, regardless of the change in the number of dragon population)

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (a - by - c)x \tag{16}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = (nx - m - p)y \tag{17}$$

There will be two situations:

1. The environmental carrying capacity cannot meet the growth needs of the dragon, that is, $a < c$ in equation (16) and $nx - m - p < 0$ in equation (17). At this time, A will decrease in quantity at a slower and slower speed until extinction, and B will also face extinction.(in this process, if either A or B dies out first, the c and p in equation (16) and equation (17) will increase correspondingly (depending on the predation of the dragon).)

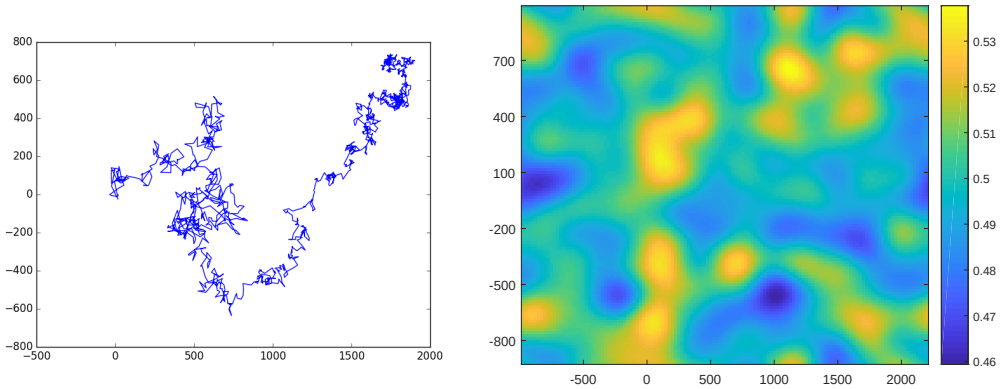
2. Environmental capacity can support the growth of the dragon, the equivalent of in equation (4) and equation (5), will decrease a and increase m, at this point A and B number was still above similar cyclical change, but the overall produce translation (translational direction according to the condition of feeding of the dragon, that is, the relative size of c and p in the equation (16) and equation (17).)

3.9 Migration Model

A program simulation is used to simulate the migration of dragons. The simulation program depends on the establishment of the model. It follows the diffusion path of finance by adjusting its movements according to its environment and energy density, while its interactions are determined by internal properties such as distance and speed.

After several simulations, we obtain the trajectories of several dragons. The confidence interval of 95% was calculated by statistical model as the survival area of the dragon.

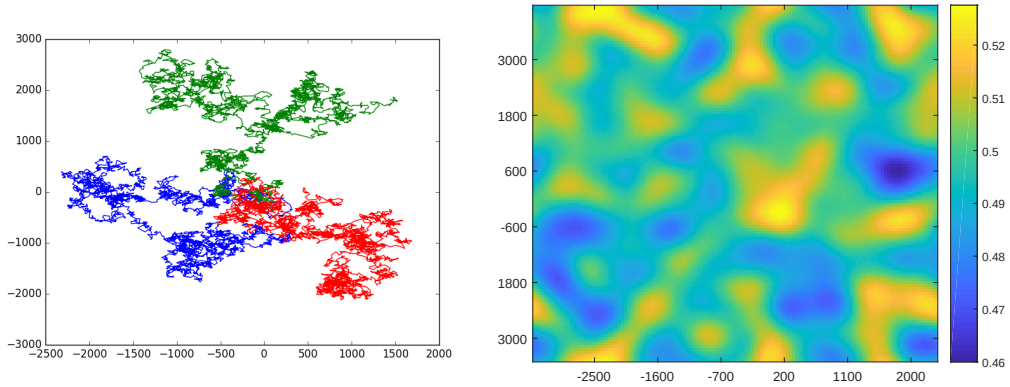
Finally,we obtain that the living space a dragon needs is **824**km^2/day



(a) The 100-year migration route

(b) Environmental energy density for one year

Figure 6: For one dragon



(a) The 10-year migration route for 3 dragons

(b) Environmental energy density for one year

Figure 7: For three dragons

4 Extended Model Application

4.1 Dragon Survival Needs Model(Under Different Climatic Conditions)

Next, we will answer that how large a community is necessary to support a dragon for varying levels of assistance that can be provided to the dragon.At the same time, we take climate into account.

Suppose that only 2% of the energy (GE) a dragon gets from food per day comes from plants (used only to obtain the methane that provides the basis for the fire) : based on the previous calculation, a dragon needs to eat vegetables per day

$$X_0 = \frac{2\%GE}{0.00557} \times 0.1kg$$

(we know that 100g vegetables – 13.3 kilocalorie – 0.0557MJ[5])

Take the standard body weight of the adult dragon is 8t, and solve for $X_0 = 808.26kg$.

The rest of the food for cattle, sheep and other herbivores.Assume they are all mountain goat: according to the average weight of the goat–32kg [6], 203 kilocalorie per 100g of mutton ($\approx 0.84854MJ$)

According to the previous calculation, the number of sheep the dragon needs to feed on per day is (unit: one)

$$Y_0 = 98\%GE \times \frac{100}{32 \times 1000 \times 0.84854 \times 60\%}$$

Take the standard weight of the adult dragon is 8t, to solve for $Y_0 = 13.54$. An

adult dragon approximately eats 13.54 mountain goats per day.

In the exposition of the ecological impact of dragons, it has been discussed that the environmental carrying capacity is not enough to support the survival of dragons.We will now discuss only the environmental support necessary for a dragon to live a stable life.

According to the daily activity area of the dragon mentioned above, the population density of mountain goats can be estimated to be at least

$$\rho_0 = \frac{Y_0}{B\% \times S}$$

B is the predation rate of the dragon (1-10 is preferable, because the dragon do not eat every sheep in its range and will not eat again when they are full. In this model, we take B=3)

Take the adult dragon standard weight is 8t, and get ρ_0=0.548 (one/km^2)

Now we will consider the climate change, the local community support changes that occur when dragons migrate to different areas.First, according to the description of the effect of the climate on the dragon, after reasonably modifying the coefficient, draw the corresponding contrast chart (the contrast between the image in the climate and the initial image).Then the change of population support to the dragon was discussed according to three typical climates.

The arid region:Take the standard weight of the adult dragon is 8t, to solve for $X_0 = 673.44kg$, $Y_0 = 11.02$, ρ_0=0.446 (one/km^2)

The warm temperate region: Take the standard weight of the adult dragon is 8t, to solve for $X_0 = 904.87kg$, $Y_0 = 15.08$, $\rho_0 = 0.623$

The Arctic region:Because of the diversity of life in the polar regions, it is no longer possible to assume that the dragon feeds on vegetables and goats.The plants in the polar regions are mainly lichens and mosses. It is assumed that the dragons in the arctic get cellulose by eating mosses, and the main food is cod.

- 100g moss – 8 kilocalorie – 0.03349MJ [5]
- 100g cod – 78 kilocalorie – 0.3265MJ [6]

$$X_0 = \frac{2\%GE}{0.03349} \times 0.1kg$$
$$Z_0 = 98\%GE \times \frac{0.1}{0.3265 \times 90\%}$$
$$\rho_0 = \frac{Z_0}{B\% \times S}$$

Take the standard weight of the adult dragon is 8t, to solve for $X_0 = 305.32kg$, $Z_0 = 255.92kg$, ρ_0=10.358 (kg/km^2)

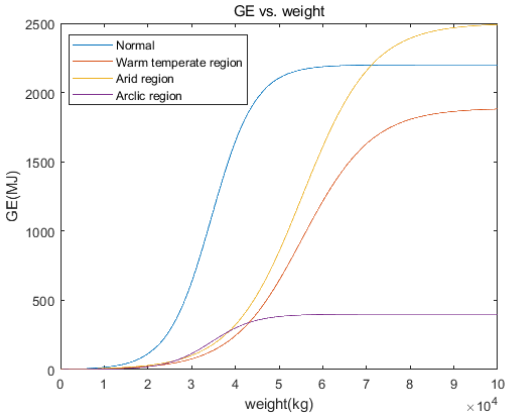


Figure 8: Weight comparison in different environments

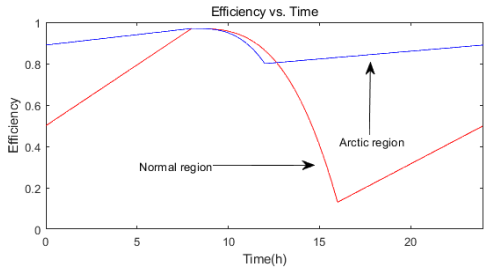
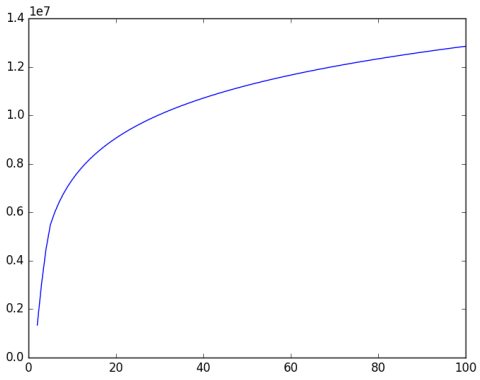


Figure 9: Efficiency in different environments

4.2 Dragon Living Space Model

Next, solve the problem of living space.Based on the model mentioned above, we can first calculate the living space required by a dragon.



The x-coordinate is the number of years and the y-coordinate is the area(km^2).

Figure 10: The range of a dragon over time

In solving the problem of dragon living space, we also innovatively explored the interaction between multiple dragons and the attraction between dragons of the opposite sex.

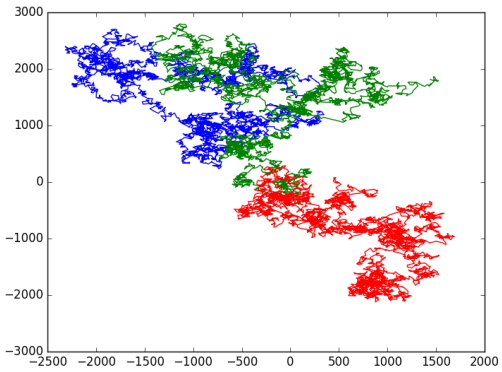


Figure 11: Interactions between dragons of the opposite sex

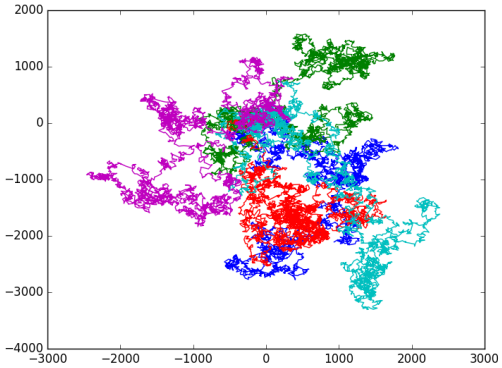


Figure 12: The interaction of five dragons

Finally,we obtain that the living spaces 3 dragon need is $2.12\times10^7 km^2/100 years$

5 Testing the Model

5.1 Sensitivity analysis

Dragon for fictional creatures, different of the film and television works books on it, so in a slightly different parameter values. Such as the weight, said a year after their birth weight in the title for 30–40 kg, and the definition of adult is not clear, so use logistics model for regression analysis of the difference is larger, the late weight may come in and go out. But they are consistent, increase first and then decreases to zero, flight properties such as energy consumption will be affected by the corresponding, but the influence of proportion was only amplified. Therefore, you can set the scaling factor to make the result consistent.As shown in figure 13.

However, if the selection of the scaling factor is inconsistent, the body weight

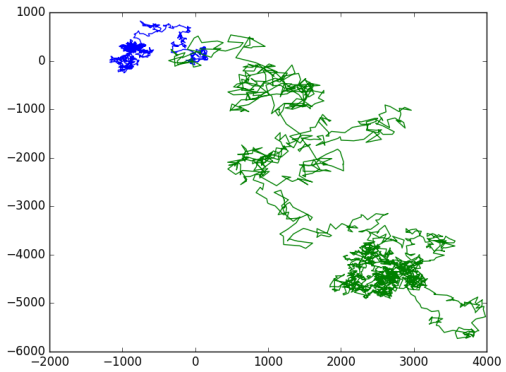


Figure 13: Sensitivity analysis

will also be affected. Thus affecting the flight speed, and the flight speed will affect the flight distance. So the range suitable for the survival of the dragon will be increased by the same proportion, and the results are different.

5.2 Stability test

To consider the actual situation of the random events, the impact of parameters using random variables instead of dragon behaviour, such as climate and energy density distribution. So the model results depends on the random number enough "random". For example, using C++ standard library random number by default engine, every time is to run the program results are the same, and the analysis of the model is not enough. The use of linear congruence, the number of seconds to will as simulated too fast lead to the change of a single period of time, is not conducive to simulate the actual situation.Therefore, in order to ensure the stability of the model, it is necessary to select an appropriate random number generation algorithm.The program USES an unsigned mason rotation generator that relies on nanoseconds.

6 Conclusions

By establishing basic models, we got a lot of valuable data, such as the ecological demand of the dragon: 13.54 sheep/day, 808.26kg of vegetables/day. The population density of sheep was $0.548/km^2$.

The ecological impact of dragons is a problem within the planning. We made assumptions about the energy demand based on the data of raptors. The region where the dragon lives can provide different levels of help to the dragon. Collect regional information, and use the method of dimensionality reduction to classify different regions. We have also expanded from one dragon to many, taking into account the impact of different climatic regions. The living space a dragon needs

is $824km^2/day$. Moreover, the interaction between dragons is studied innovatively, considering the competition between dragons and the attraction between dragons of the opposite sex.

The migration of the dragon is an important part of the survival of the species. Multivariate analysis is used to determine the ecological factors between different environments. The impact of the change can be determined by establishing a corresponding analysis between the migration of the dragon and the resources needed for its growth.

7 Strengths and Weaknesses

Strengths

1. The model is practical, the hypothesis is reasonable and the data can be checked.The hypothesis is to refer to the original work and some film and television works, and start from the animals that can be contacted in life. The data is easy to collect and conforms to the biological principle.

2. The model is comprehensive.The coupling relationship is simplified and the simulation optimization is improved as simple as possible.

3. Clean Architecture, strong expanding, strong commonality. With TDD, any development node can come up with a product that can be used, containing a few bugs, has certain functions and can be released.It was transplanted from Python to C++ in a mixed way, without loss of extensibility and obvious efficiency. It simulated three dragons in 100 years in less than one minute.

Weaknesses

1. The model has made reasonable approximation in large scale for efficiency, but the simulation will cause large deviation after more than 100 years. And the API needs to be modified again to achieve the required accuracy.

MEMORANDUM

To: Mr. George R. R. Martin

From: Dragon lovers

Dear Mr. George R.R. Martin,

Thank you for this opportunity to write to express our love for your work and our thoughts on the dragon in the book. We learned that your novel *A Song of Ice and Fire* is set in medieval Europe, so we have a bold idea: if you put your story on the stage today, the dragon in the book will live in the current environment, what kind of impact it will have. We've come up with some analysis that we hope will help you maintain a realistic ecological basis for your story.

As far as we know, there has never been anything in nature like what we define as "dragon" today. We guess your question might be: "the dragon's flame how to realize", "dragon daily activity and the energy consumption", "the movement of dragons from arid regions to temperate regions and arctic regions".

In books, dragons are fire-breathing animals, and we can't get a direct reference from reality. Currently, there is no known animal in nature can spray fire, some of the so-called "breath fire" are just fluorescent chemicals. We take inspiration from the rumen digestion of ruminants: ruminants such as cattle and sheep can convert cellulose to methane through the rumen, even though the methane produced by these animals is excreted. In our scenario, dragons have a digestive mechanism similar to that of ruminants, producing methane and storing it in an organ of the body.(for this reason, the dragon's diet should include plants to get the cellulose needed to break down methane.) When the dragon needs to breathe fire, the methane stored in its body is transported to its lower jaw, where it is exhaled through friction and high-speed exhalation, and then burned in the air, creating a jet of flame. Therefore, you should add some vegetables to the dragon's diet in your novel, except lambs and calves.

The daily life of an animal is closely related to its energy consumption, and when the dragon engages in war without being influenced by human factors (as described in the novel), its daily activities are similar to those of other animals in nature, from which we can draw a reference. Some lazy animals in nature, such as the koala, spend only four hours a day on foraging, moving, cleaning and interacting with other koalas, and the rest of the day is for sleeping in trees. This is because their food is of low nutritional value and they cannot get enough energy. They can only reduce their daily energy consumption by increasing their sleep time. Large herbivores, on the other hand, must spend a lot of time eating and sleeping in order to get the energy they need to maintain their lives.

For the dragon, as a huge animal, its energy consumption is also huge, so it has to absorb a lot of food. Through analysis, we obtained a graph of the dragon's

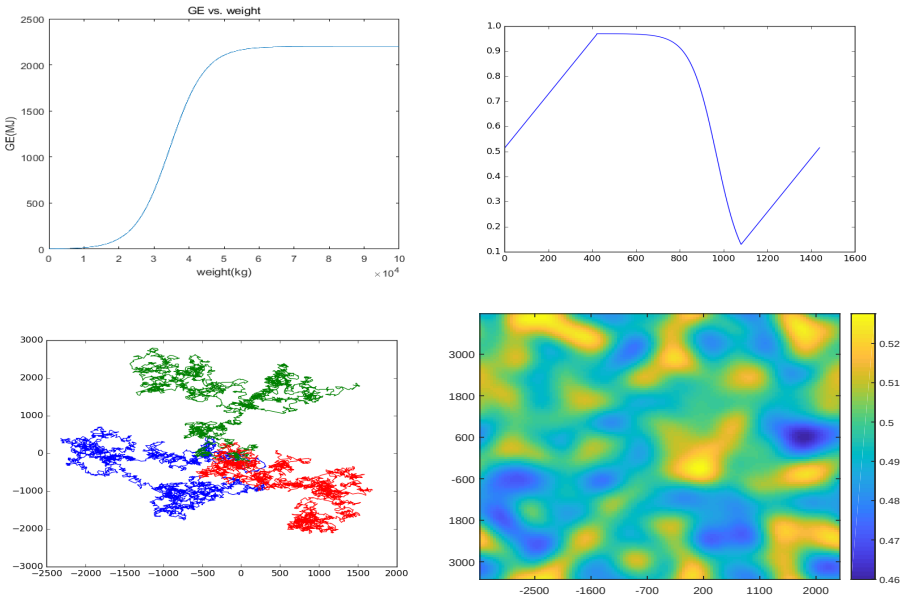
food intake. It has been estimated that an adult dragon weighing about 8t would need to eat about 13.5 sheep and 808.26kg of vegetables a day. So in your book, dragons can no longer be described as unbridled gluttons, they should eat a balanced diet every day, reasonable distribution of nutrition. In order to describe the daily activities of the dragon, we also introduced the concept of mental state, which describes the changes of the dragon's energy with time. The sleep of the dragon will supplement it every night, and it will slowly wear out during the daily activities. As you can see from the picture, it describes the general activity of the dragon during the day. For reference, you can set the schedule for the dragon in the book.

Like many birds, dragons will migrate. But unlike other birds that migrate to adapt to climate, we think that because of the spirituality of the dragon, its migration model is essentially random. Because of its extremely fast speed and long flying range, the dragon covered a very wide area during its migration. So it moves from arid regions to temperate regions, to the arctic. But at the same time we consider the influence of environmental energy density and interaction between dragons in this model. Through the program simulation, the migration route of the dragon is obtained. During the migration process of the dragon, it will reach the environment with different climatic conditions due to the randomness of movement. However, due to the limitation of environmental tolerance, it will spend less time in the harsh environment and more time in the comfortable environment. A dragon is never an animal that stays put. You can better describe the migration of a dragon based on the picture.

Thank you again for taking the time to read our suggestions. We sincerely hope that the models we built can help with maintaining the realistic ecological underpinning of the story!

Sincerely,

Dragon lovers



References

[1] Game of Thrones Wiki. [Online] Available: https://gameofthrones.fandom.com/wiki/Dragons

[2] Yunqing, Y., & Yulien, Y. (1994). ROUTINE PARAMETER ESTIMATION METHODS FOR ANIMAL GROWTH MODELS [J]. Journal of Northeast Agricultural University, 2.

[3] DECD, O. (1991). Estimation of greenhouses gas emissions and sinks. *Report presented to IPCC*.

[4] Hongmin, D., Erda, L., & Qichang, Y. (1995). Methane emitted from ruminants in China and the mitigation technologies.[J]. *Rural Eco-Environment*, *3*.

[5] Food calorie inquiry table Boohee[Online] Available: `http://www.boohee.com/food/`.

[6] "mountain goats" Encyclopedia of Life[Online]. Available: `https://eol.org/`.

Simulation

Here are simulation programs we used in our model as follow.

```
all:
    cython3 utils/plot_utils.pyx
    g++ -o main main.cpp -std=c++11 -O -I /usr/include/python3.5/ -l python3.5m
clean:
    rm main
```

```
/*
 * @Time     : 2019/01/25 13:00
 * @Author   : Iydon
 * @File     : main.cpp
 */

#include <stdio.h>
#include "time.h"
#include "environ.h"
#include "dragon.h"
#include "interface.h"
#include "utils/plot.h"
using namespace std;

int main() {
    int YEAR = 1000;
    int DAY  = 365;
    int MIN  = 60 * 24;
    double DELTA = 1.0 / DAY / MIN;
    Time time = Time(0, 0, 0);
    int y, d, m;

    Dragon Smaug = Dragon(1);
    Smaug.set_attributions(0, true, 10.0, 10.0, 1.0, 0.0, 0.0, 1.0);

    Environ dessert = Environ(0);
    dessert.set_attributions(27.0, 0.7, 1000.0);

    double xx[YEAR*DAY];
    double yy[YEAR*DAY];

    for (y=0; y<YEAR; y++) {
        for (d=0; d<DAY; d++) {
            for (m=0; m<MIN; m++) {
                time.set_minute(m);
                Smaug.simulate(DELTA, time, dessert);
                dessert.simulate(time);
            }
            xx[y*DAY+d] = Smaug.get_x();
            yy[y*DAY+d] = Smaug.get_y();
            time.set_day(d);
        }
        time.set_year(y);
    }

```

```
    plot(xx, yy, YEAR*DAY, "result.png");

    Smaug.print_info();
}
```

```
/*
 * @Time     : 2019/01/27 19:20
 * @Author   : Iydon
 * @File     : time.h
 */

class Time {
    public:
        Time(int,int,int);
        void set_year(int);
        void set_day(int);
        void set_minute(int);
        int get_year(void) { return _year; }
        int get_day(void) { return _day; }
        int get_minute(void) { return _minute; }
        double double_in_years(void);
        double double_in_day(void);
    private:
        int _year, _day, _minute;
        int _YEAR2DAY   = 365;
        int _DAY2MINUTE = 60 * 24;
};

Time::Time(int year, int day, int minute) {
    _year   = year;
    _day    = day;
    _minute = minute;
}

void Time::set_year(int year) {
    _year = year;
}

void Time::set_day(int day) {
    _day = day;
}

void Time::set_minute(int minute) {
    _minute = minute;
}

double Time::double_in_years(void) {
    return (double)_year + double_in_day()/_YEAR2DAY;
}

double Time::double_in_day(void) {
    return (double)_day + (double)_minute/_DAY2MINUTE;
}
```

```
/*
```

```
 * @Time     : 2019/01/25 13:10
 * @Author   : Iydon
 * @File     : environ.h
 */

#include <stdio.h>


class Environ {
    public:
        /* functions */
        Environ(int);
        void set_attributions(double,double,double);
        void set_temperature(double);
        void set_humidity(double);
        void set_energy_dens(double);
        void add_temperature(double);
        void add_humidity(double);
        void add_energy_dens(double);
        double get_temperature(void){ return _temperature; }
        double get_humidity(void){ return _humidity; }
        double get_energy_dens(void){ return _energy_dens; }
        void print_info(void);
        /* interface */
        void simulate(Time);
        /* rules */
    private:
        /* attributions */
        unsigned int _id    = 0;
        bool   _display     = true;
        double _temperature = 0.0;
        double _humidity    = 0.0;
        double _energy_dens = 0.0;
        /* part constant */
        double _period = 1.0;
};

Environ::Environ(int id) {
    _id = id;
}

void Environ::set_attributions(double temperature, double humidity, double energy_den
    _temperature = temperature;
    _humidity    = humidity;
    _energy_dens = energy_dens;
    print_info();
}

void Environ::set_temperature(double value) {
    _temperature = value;
    if (_temperature<-273.15) _temperature=-273.15;
}

void Environ::set_humidity(double value) {
    _humidity = value;
```

```
    if (_humidity<0.0) _humidity=0.0;
    else if (_humidity>1.0) _humidity=1.0;
}

void Environ::set_energy_dens(double value) {
    _energy_dens = value;
    if (_energy_dens<0.0) _energy_dens=0.0;
}

void Environ::add_temperature(double value) {
    _temperature += value;
    if (_temperature<-273.15) _temperature=-273.15;
}

void Environ::add_humidity(double value) {
    _humidity += value;
    if (_humidity<0.0) _humidity=0.0;
    else if (_humidity>1.0) _humidity=1.0;
}

void Environ::add_energy_dens(double value) {
    _energy_dens += value;
    if (_energy_dens<0.0) _energy_dens=0.0;
}

void Environ::print_info(void) {
    if (_display) {
        printf("%10s: %10x\n", "ID", _id);
        printf("%10s: %10.3f \n", "Tempature", _temperature);
        printf("%10s: %10.3f %%\n", "Humidity", _humidity*100);
        printf("%10s: %10.3f kJ/mš\n", "Energy ", _energy_dens);
        printf("\n");
    }
}
```

```
/*
 * @Time     : 2019/01/25 13:10
 * @Author   : Iydon
 * @File     : dragon.h
 */

#include <stdio.h>


class Dragon {
    public:
        /* functions */
        Dragon(int);
        void set_attributions(double,bool,double,double,double,double,double,double);
        void set_weight(double);
        void set_energy(double);
        void set_spirit(double);
        void set_x(double);
        void set_y(double);
        void set_v(double);
```

```
21          void add_weight(double);
22          void add_energy(double);
23          void add_spirit(double);
24          void add_x(double);
25          void add_y(double);
26          void add_v(double);
27          void print_info(void);
28          double get_weight(void) { return _weight; }
29          double get_energy(void) { return _energy; }
30          double get_spirit(void) { return _spirit; }
31          double get_x(void) { return _x; }
32          double get_y(void) { return _y; }
33          /* interface */
34          void simulate(double,Time,Environ);
35          void grow_up(Time,double);
36          void move_on(Time,Environ);
37          void set_status(int,int);
38          int free(Time,Environ);
39          int flying(Time,Environ);
40          int hunting(Time,Environ);
41          int sleeping(Time,Environ);
42          void interaction(Dragon);
43          double satisf_index(Environ);
44      private:
45          /* attributions */
46          unsigned int _id  = 0;
47          double _age       = 0;
48          bool   _sex       = true;
49          double _weight    = 10.0;
50          double _energy    = 10.0;
51          double _spirit    = 1.0;
52          double _x=0.0, _y=0.0, _v=1.0;
53          /* part constant */
54          double _max_energ = _energy;
55          /* flag */
56          unsigned int _is_free     = 1;
57          unsigned int _is_flying   = 0;
58          unsigned int _is_hunting  = 0;
59          unsigned int _is_sleeping = 0;
60          /* count */
61          unsigned long int _count_free     = 0;
62          unsigned long int _count_flying   = 0;
63          unsigned long int _count_hunting  = 0;
64          unsigned long int _count_sleeping = 0;
65  };
66
67  Dragon::Dragon(int id) {
68      _id = id;
69  }
70
71  void Dragon::set_attributions(double age, bool sex, double weight, double energy,
72                                double spirit, double x, double y, double v) {
73      _age    = age;
74      _sex    = sex;
75      _weight = weight;
```

```
76      _energy = energy;
77      _spirit = spirit;
78      _x      = x;
79      _y      = y;
80      _v      = v;
81      print_info();
82  }
83
84  void Dragon::set_weight(double value) {
85      _weight = value;
86      if (_weight<0.0) _weight=0.0;
87  }
88
89  void Dragon::set_energy(double value) {
90      _energy = value;
91      if (_energy<0.0) _energy=0.0;
92  }
93
94  void Dragon::set_spirit(double value) {
95      _spirit = value;
96      if (_spirit<0.0) _spirit=0.0;
97      else if (_spirit>1.0) _spirit=1.0;
98  }
99
100 void Dragon::set_x(double value) {
101     _x = value;
102 }
103
104 void Dragon::set_y(double value) {
105     _y = value;
106 }
107
108 void Dragon::set_v(double value) {
109     _v = value;
110     if (_v<0.0) _v=0.0;
111 }
112
113 void Dragon::add_weight(double value) {
114     _weight += value;
115     if (_weight<0.0) _weight=0.0;
116 }
117
118 void Dragon::add_energy(double value) {
119     _energy += value;
120     if (_energy<0.0) _energy=0.0;
121 }
122
123 void Dragon::add_spirit(double value) {
124     _spirit += value;
125     if (_spirit<0.0) _spirit=0.0;
126     else if (_spirit>1.0) _spirit=1.0;
127 }
128
129 void Dragon::add_x(double value) {
130     _x += value;
```

```
131 }
132
133 void Dragon::add_y(double value) {
134     _y += value;
135 }
136
137 void Dragon::add_v(double value) {
138     _v += value;
139     if (_v<0.0) _v=0.0;
140 }
141
142 void Dragon::set_status(int status, int num) {
143     /*
144      * 0: Free.
145      * 1: Flying.
146      * 2: Hunting.
147      * 3: Sleeping.
148      */
149     _is_free     = 0;
150     _is_flying   = 0;
151     _is_hunting  = 0;
152     _is_sleeping = 0;
153     switch (status) {
154         case 0: {
155             _is_free = num;
156             break;
157         }
158         case 1: {
159             _is_flying = num;
160             break;
161         }
162         case 2: {
163             _is_hunting = num;
164             break;
165         }
166         case 3: {
167             _is_sleeping = num;
168             break;
169         }
170     }
171 }
172
173 void Dragon::print_info(void) {
174     printf("%10s: %10x\n", "ID", _id);
175     printf("%10s: %10.1f year(s)\n", "Age", _age);
176     printf("%10s: %10s\n", "Sex", _sex?"Male":"Female");
177     printf("%10s: %10.1f kg\n", "Weight", _weight);
178     printf("%10s: %10.1f kJ\n", "Energy", _energy);
179     printf("%10s: %10.1f %%\n", "Spirit", _spirit*100);
180     printf("%10s: %10.1f km\n", "Coor X", _x);
181     printf("%10s: %10.1f km\n", "Coor Y", _y);
182     printf("%10s: %10.1f km/min\n", "Velocity", _v);
183     printf("%10s: %10.1f km/h\n", "Velocity", _v*60);
184     printf("\n");
185     printf("%10s %10lu minute(s)\n", "Free", _count_free);
```

```
186     printf("%10s %10lu minute(s)\n", "Flying", _count_flying);
187     printf("%10s %10lu minute(s)\n", "Hunting", _count_hunting);
188     printf("%10s %10lu minute(s)\n", "Sleeping", _count_sleeping);
189     printf("\n");
190 }
```

```
1  /*
2   * @Time     : 2019/01/25 13:18
3   * @Author   : Iydon
4   * @File     : interface.h
5   */
6
7  #include <random>
8  #include <math.h>
9
10 using namespace std;
11
12
13 std::default_random_engine e;
14
15
16 /*
17  * Dragon.
18  */
19 void Dragon::simulate(double inc, Time time, Environ envir) {
20     grow_up(time, inc);
21     move_on(time, envir);
22 }
23
24  void Dragon::grow_up(Time time, double inc) {
25     /*
26      * time, spirit, energy.
27      */
28     _age += inc;
29     if (time.get_minute()==0) {
30         add_weight(-7500.0/(1+596.0*exp(-_age))+7500.0/(1+596.0*exp(-_age-1.0/365)));
31     }
32     if (_is_sleeping==0) {
33         set_spirit(0.970284649/(1.0+0.000000098*exp((double)time.get_minute()/60)));
34     }
35     /* unsolved */
36     lognormal_distribution<> ln_v(-_age*10000, 0.01);
37     lognormal_distribution<> ln_e(-_age, 1);
38     add_v(ln_v(e)/50);
39     add_energy(ln_e(e));
40     _max_energ = _energy;
41  }
42
43  void Dragon::move_on(Time time, Environ env) {
44     /*
45      * hunting, sleeping, flying, free.
46      */
47     if (_is_free==0) {
48         if (_is_flying) flying(time, env);
49         else if (_is_hunting) hunting(time, env);
```

```
50          else if (_is_sleeping) sleeping(time, env);
51          else set_status(0, free(time, env)+1);
52          return;
53      }
54      uniform_real_distribution<> u(0, 1);
55      if (_spirit<0.13) {
56          set_status(3, sleeping(time, env)+1);
57          return;
58      }
59      if (_energy<10.0) {
60          set_status(2, hunting(time, env)+1);
61          return;
62      }
63      if (u(e)<0.7) {
64          set_status(1, flying(time, env)+1);
65          return;
66      }
67      free(time, env);
68  }
69
70  int Dragon::free(Time time, Environ env) {
71      _count_free += 1;
72      _is_free -= 1;
73      return 10;
74  }
75
76  int Dragon::flying(Time time, Environ env) {
77      /*
78       * ed:
79       *     None.
80       * ing:
81       *     energy, spirit.
82       */
83      _count_flying += 1;
84      _is_flying -= 1;
85      uniform_real_distribution<> u(-1, 1);
86      add_x(u(e)*_v);
87      add_y(u(e)*_v);
88      add_weight(-_v/10);
89      add_energy(-_v);
90      add_spirit(-1.0/_v);
91      return 10;
92  }
93
94  int Dragon::hunting(Time time, Environ env) {
95      /*
96       * ed:
97       *     energy.
98       * ing:
99       *     weight, spirit, energy.
100      */
101     _count_hunting += 1;
102     _is_hunting -= 1;
103     add_weight(_max_energ/10000);
104     set_energy(_max_energ);
```

```
    add_spirit(-1.0/_v);
    return 10;
 }

 int Dragon::sleeping(Time time, Environ env) {
    /*
     * ed:
     *     spirit.
     * ing:
     *     spirit.
     */
    uniform_int_distribution<> u(-60,60);
    _count_sleeping += 1;
    _is_sleeping -= 1;
    add_spirit(0.001077);
    return 720 + u(e);
 }

void interaction(Dragon dragon) {
    /* too ugly to show */
}

double Dragon::satisf_index(Environ env) {
    return env.get_energy_dens()
         - abs(env.get_temperature()-27.0)
         - 10*abs(env.get_humidity()-0.7);
}


/*
 * Environment.
 */
void Environ::simulate(Time time) {
    normal_distribution<> n_t(0, 0.1);
    normal_distribution<> n_h(0, 0.001);
    normal_distribution<> n_e(0, 1);
    add_temperature(0.01*sin(2*3.1415927*time.double_in_years()) + n_t(e));
    add_humidity(n_h(e));
    if (_temperature>30.0) {
        add_energy_dens(10.0 + n_e(e));
    }
    if (_humidity>0.5) {
        add_energy_dens(10.0 + n_e(e));
    }
}
```

```
# -*- coding: utf-8 -*-
# @Time     : 2019/01/26 18:10
# @Author   : Iydon
# @File     : plot_utils.pyx
# @Compiler : Cython

import matplotlib.pyplot as plt


```

```
cdef public plot(x, y, name):
    plt.plot(x, y)
    plt.savefig(name)

cdef public loglog(x, y, name):
    plt.loglog(x, y)
    plt.savefig(name)
```

```
#include <Python.h>
#include "plot_utils.h"
#include "plot_utils.c"

void plot(double *x, double *y, int length, const char* name) {
    Py_Initialize();
    PyInit_plot_utils();

    PyObject *_x = PyList_New(length);
    PyObject *_y = PyList_New(length);

    for (int i=0; i<length; i++) {
        PyList_SetItem(_x, i, PyFloat_FromDouble(x[i]));
        PyList_SetItem(_y, i, PyFloat_FromDouble(y[i]));
    }

    plot(_x, _y, PyUnicode_FromString(name));

    Py_Finalize();
}

void loglog(double *x, double *y, int length, const char* name) {
    Py_Initialize();
    PyInit_plot_utils();

    PyObject *_x = PyList_New(length);
    PyObject *_y = PyList_New(length);

    for (int i=0; i<length; i++) {
        PyList_SetItem(_x, i, PyFloat_FromDouble(x[i]));
        PyList_SetItem(_y, i, PyFloat_FromDouble(y[i]));
    }

    loglog(_x, _y, PyUnicode_FromString(name));

    Py_Finalize();
}
```

Curve fitting

Here are curve fitting programs we used in our model as follow.

```
warning off;
when_to_adult = 20;
```

```
data = [0,10; 1,35; when_to_adult,7500];
x = data(:, 1);
y = data(:, 2);

%% Brody
ft = fittype( 'exp1' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.StartPoint = [0 0];
[fitresult, gof] = fit_plot(x, y, ft, opts, 'Brody-1'); %#ok

ft = fittype({'1', '(exp(-x))'}, 'independent', 'x', 'dependent', 'y', 'coefficients'
opts = fitoptions;
[fitresult, gof] = fit_plot(x, y, ft, opts, 'Brody-2'); %#ok


%% Logistic
ft = fittype('a/(1+b*exp(-x))', 'independent', 'x', 'dependent', 'y');
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.StartPoint = rand(1, 2);
[fitresult, gof] = fit_plot(x, y, ft, opts, 'Logistic'); %#ok


%% Bertalanffy
ft = fittype( '[a/b-(a/b-10^(1/3))*exp(-b*x/3)]^3', 'independent', 'x', 'dependent',
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.StartPoint = rand(1, 2);
[fitresult, gof] = fit_plot(x, y, ft, opts, 'Bertalanffy'); %#ok


%% Gompertz
ft = fittype( 'a*exp(-exp(b-c*x))', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.StartPoint = rand(1, 3);
[fitresult, gof] = fit_plot(x, y, ft, opts, 'Gompertz'); %#ok


%% Richards
ft = fittype( 'a*(1-b*exp(-x))^c', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.StartPoint = rand(1, 3);
[fitresult, gof] = fit_plot(x, y, ft, opts, 'Richards'); %#ok


%% Janoschek
ft = fittype( 'a-(a-10)*exp(-b*x)', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Algorithm = 'Levenberg-Marquardt';
opts.Display = 'Off';
opts.StartPoint = rand(1, 2);
[fitresult, gof] = fit_plot(x, y, ft, opts, 'Janoschek');
```

```
58
59
60
61 %% Fit function
62 function [fitresult, gof] = fit_plot(x, y, ft, opts, name)
63 %  Args:
64 %      X Input : x
65 %      Y Output: y
66 %  Output:
67 %      fitresult : a fit object representing the fit.
68 %      gof : structure with goodness-of fit info.
69     try
70         [fitresult, gof] = fit(x, y, ft, opts);
71         model = struct(fitresult);
72         % model.defn and model.coeffValues
73         disp([name, '    RMSE: ', num2str(gof.rmse), ...
74                     '    R^2: ', num2str(gof.rsquare)]);
75         figure('Name', name);
76         hold on;
77         loglog(x, y, 'o');
78         xx = linspace(0, 100, 128);
79         loglog(xx, fitresult(xx));
80         legend('Y vs. X', name, 'Location', "NorthEast");
81         text(50, 3000, ['R^2: ', num2str(gof.rsquare)])
82         title([name, ': ', model.defn]);
83         xlabel Time(Year)
84         ylabel Weight(kg)
85         grid on;
86         saveas(gcf,name,'epsc')
87     catch
88         disp([name, ' Failed']);
89         fitresult = NaN;
90         gof = NaN;
91     end
92 end
```