目录

[一、 错误修正和更新 2](#_Toc127716138)

[1.1 错误修正 2](#_Toc127716139)

[1.2 更新 2](#_Toc127716140)

[二、 问题1：干旱条件的物种竞争模型 3](#_Toc127716141)

[2.1 模型建立 3](#_Toc127716142)

[2.1.1 两物种竞争模型 3](#_Toc127716143)

[2.1.2 多物种竞争模型 6](#_Toc127716144)

[2.1.3 干旱条件下多物种竞争模型 6](#_Toc127716145)

[2.2 模型求解 8](#_Toc127716146)

[2.2.1 生长率与降水量关系 8](#_Toc127716147)

[2.2.2 微分方程参数设置 9](#_Toc127716148)

[2.2.3 改进欧拉法 10](#_Toc127716149)

[2.2.4 求解结果 14](#_Toc127716150)

[三、 问题2与问题3：物种数量最优控制模型 14](#_Toc127716151)

[3.1 问题分析 15](#_Toc127716152)

[3.2 模型建立 15](#_Toc127716153)

[3.2.1 决策变量 15](#_Toc127716154)

[3.2.2 目标函数 15](#_Toc127716155)

[3.2.3 控制方程 16](#_Toc127716156)

[3.2.4 约束条件 16](#_Toc127716157)

[3.2.5 初始条件 16](#_Toc127716158)

[3.2.6 总体模型 16](#_Toc127716159)

[3.3 模型求解 17](#_Toc127716160)

[3.3.1 处理约束条件：罚函数法 17](#_Toc127716161)

[3.3.2 遗传算法 21](#_Toc127716162)

[3.3.3 遗传算法的伪代码 22](#_Toc127716163)

[3.3.4 多变异位自适应遗传算法 23](#_Toc127716164)

[3.3.5 整体的优化问题求解 23](#_Toc127716165)

[3.4 求解结果 24](#_Toc127716166)

[四、 问题4 26](#_Toc127716167)

[五、 问题5 26](#_Toc127716168)

[六、 问题6 26](#_Toc127716169)

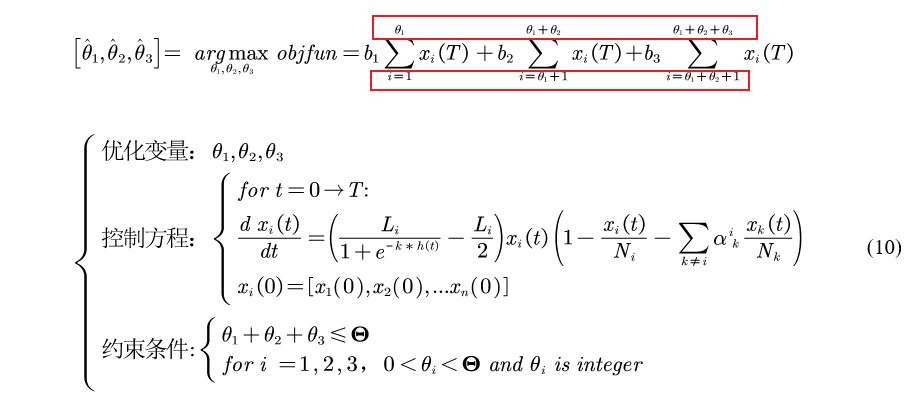
[七、 灵敏性分析 27](#_Toc127716170)

文档更新时间：2023.2.19 16:00，后续更新地址<https://mbd.pub/o/bread/ZJWWkply>

# 错误修正和更新

## 错误修正

**修正3.0版本求和上下标问题**



## 更新

更新了3.4节

**问题A：受干旱影响的植物群落**

****

**背景介绍**

不同种类的植物以不同的方式对压力作出反应。例如，草场对干旱相当敏感。干旱发生的频率和严重程度各不相同。

许多观察表明，不同物种的数量对植物群落在连续几代人暴露于干旱周期时的适应性起作用。在一些只有一种植物的群落中，随后的几代人对干旱条件的适应性不如有四个或更多物种的群落中的单个植物。这些观察结果提出了许多问题。例如，一个植物群落从这种局部生物多样性中受益所需的最小物种数量是多少？随着物种数量的增加，这种现象是如何扩展的？这对一个植物群落的长期生存能力意味着什么？

**要求**

考虑到干旱适应性与植物群落中物种数量的关系，你的任务是探索并更好地理解这一现象。具体来说，你应该

# 问题1：干旱条件的物种竞争模型

建立一个数学模型，预测一个植物群落在各种不规则的天气周期中如何随时间变化。包括本该降水充足的干旱时期。该模型应考虑到干旱周期中不同物种之间的相互作用。

## 模型建立

**看到群落一词，我们建立物种竞争模型，并加入环境因素（降水量）**

### 两物种竞争模型

**首先简单起见，先建立2生物，无环境因素的竞争模型：**

研究在同一个自然环境中生存的两个种群之间的竞争关系。假设两个种群独自在这个自然环境中生存时数量演变都服从**Logistic规律**，又假设当它们相互竞争时都会减慢对方数量的增长，增长速度的减小都与它们数量的乘积成正比。按照这样的假设建立的常微分方程模型为



方程的另一种形式为：



其中r为增长率，N为物种最大容量，

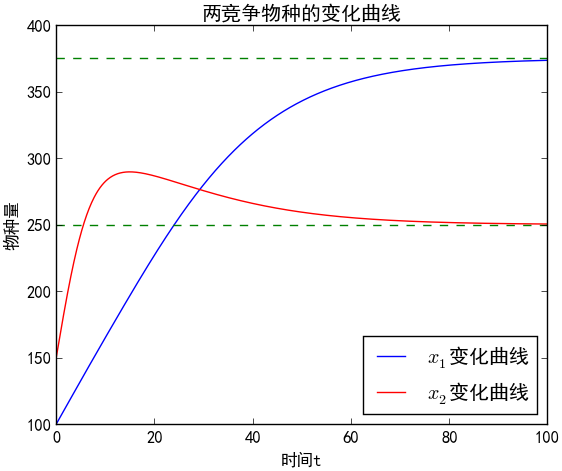
α：物种2对物种1的竞争系数，即每个N2个体所占用的空间相当于α个N1个体所占用空间。

β：物种1对物种2的竞争系数，即每个N1个体所占用的空间相当于β个N2个体所占用空间。

在分析的过程中，我们所采用的模型参数是



这些参数表明， $x\_1$ 是一个增长相对缓慢的物种，但是种内竞争较小，而且不容易受到 $x\_2$ 物种的影 响，因此，长远来看， $x\_1$ 物种将在该自然环境中占优；相比之下， $x\_2$ 物种的增长速度较快，但是种 内竞争较大，而且容易受到 $x\_1$ 物种的影响，因此在长期的竞争中， $x\_2$ 物种将处于稍弱的地位。下面 的数值求解结果支持了这一推测。



数值解表明，物种开始增长较快，达到最大值后开始缓慢下降，最后稳定在250；而则保持相对较慢的增长速度，直到平衡点375。

代码如下：

|  |
| --- |
| from scipy.integrate import odeint  import numpy as np  from scipy.optimize import leastsq  import matplotlib.pyplot as plt  plt.rcParams['font.sans-serif'] = ['SimHei'] #这两句用来正常显示中文标签  plt.rcParams['axes.unicode\_minus'] = False  t = np.arange(0,100,0.1)  def deriv(w,t,a,b,c,d,e,f):  x,y = w  return np.array([ a\*(1-b\*x)\*x-c\*y\*x, d\*(1-e\*y)\*y-f\*x\*y])  p=[0.1, 0.002, 0.0001, 0.3, 0.003, 0.0002, 100, 150]  a,b,c,d,e,f,x0,y0=p  yinit = np.array([x0,y0]) # 初值  yyy = odeint(deriv,yinit,t,args=(a,b,c,d,e,f))  plt.figure(figsize=(7,5))  plt.plot(t,yyy[:,0],"b-",label="$x\_1$变化曲线")  plt.plot(t,yyy[:,1],"r-",label="$x\_2$变化曲线")  plt.plot([0,100],[250,250],"g--")  plt.plot([0,100],[375,375],"g--")  plt.xlabel(u'时间t')  plt.ylabel(u'物种量')  plt.title(u'两竞争物种的变化曲线')  plt.legend(loc=4)  plt.show() |

### 多物种竞争模型

**现在将模型扩展一下，建立多物种竞争模型，也就是群落的竞争模型，**

将公式2扩展一下



其中为增长率，为物种最大容量，

:物种k对物种i的竞争系数，即每个Nk个体所占用的空间相当于个Ni个体所占用空间。

### 干旱条件下多物种竞争模型

**然后我们再加入加入环境因素（降水量）的影响：**

我们知道，降水量（符号h表示降水量）应该主要影响的是生长率，所以主要任务是建立以降水量h为自变量，生长率为因变量的函数。

以降水量h为自变量，生长率为因变量的函数可能是以下形式：

（1）生长率与降水量h成正比：即



注意由于题目要求：**在各种不规则的天气周期中如何随时间变化**，包括本该降水充足的干旱时期。所以这里降水量h应该和时间有关，生长率也和时间有关。在求解模型的时候，由于题目没给降水量随时间的曲线，**需要我们自己生成一些降水量随时间的曲线数据，比如说前一半时间降水多，后一半时间降水少，或者前一半时间降水少，后一半时间降水少等等。**

（2）生长率与降水量h成**logistic关系，这一关系的原因简单解释一下：降水量多的时候植物可能会淹死，因此生长率不可能无限增大，这一点需要写作时在假设一节详细解释原因！**

逻辑斯谛函数（英语：logistic function）是一种常见的S型函数，其函数图像称为逻辑斯谛曲线（英语：logistic curve）。简单的逻辑斯谛函数可用下式表示：



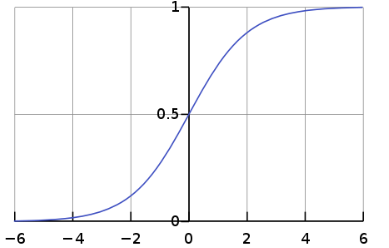
其中 :

为 S 形曲线中点的 x 值；

L 为曲线的最大值

k 为逻辑斯谛增长率或曲线的陡度

标准逻辑斯谛函数，其中



降水量为0时，假设生长率也为0，构造一个逻辑斯谛函数并减去降水量为0时的数值，就得到了降水量为0时生长率也为0的改进逻辑斯谛函数；



其中为物种i的最大生长率， 为逻辑斯谛增长率或曲线的陡度，h(t)为降水量。

将上式代入式，并把都写成t的函数：



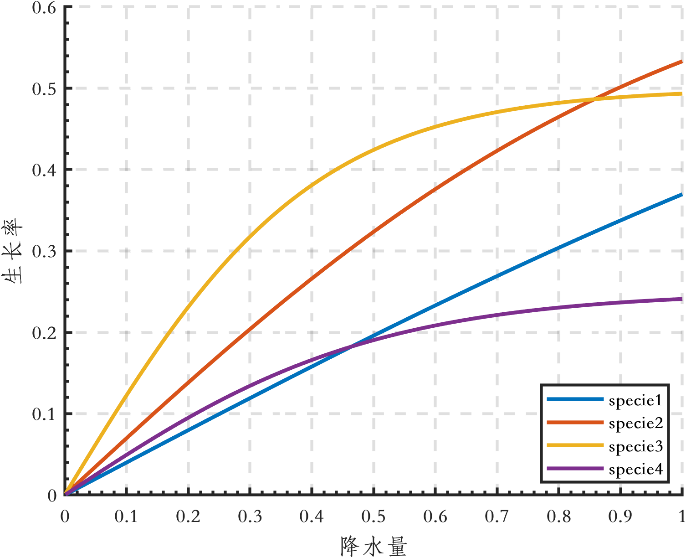
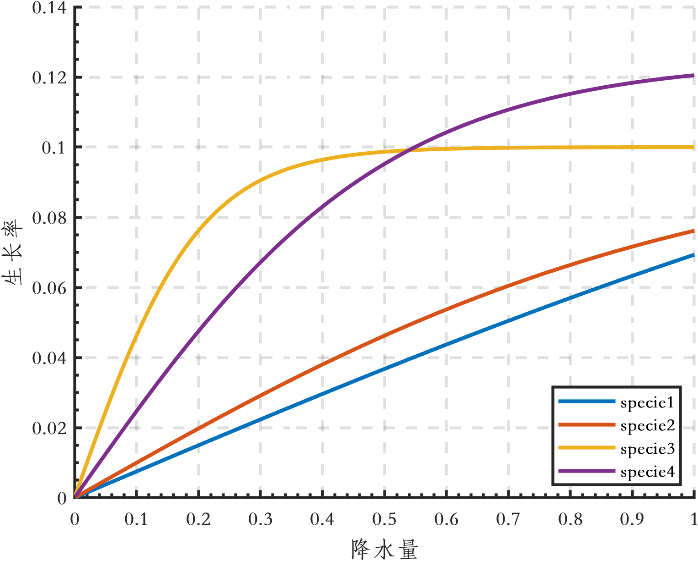
**然后我们就可以自行构造天气条件（降水量），假设物种数量（比如3种 or 4种），设置相关参数，求解微分方程模型，画出各物种的数量变化曲线。然后就能进行各种分析（语文建模）**

## 模型求解

### 生长率与降水量关系

先来探究多物种情况下，生长率关于降水量的函数图像

|  |
| --- |
| 多物种情况下，生长率关于降水量的函数图像的代码（不可直接运行，运行代码请打开problem1.m文件） |
| clc**;** clear**;** close all**;**  year**=**2**;**%模拟年数  T**=**360**\***year**;**%方便起见，一年设置为360天，一个季度90天  dt**=**1**;**%时间间隔：1天  tlen**=**T**/**dt**;**  tspan **=** linspace**(**1**,** T**,** tlen**);**  n**=**4**;**%物种数量  %%  L**=[**0.3**,**0.2**,**0.2**,**0.25**];**%物种i的最大生长率  k**=[**1**,**2**,**10**,**4**];**%物种i的逻辑斯谛增长率  h**=(**1**/**tlen**)\*(**1**:**tlen**);**%归一到0-1的降水量h(t)  r**=**zeros**(**n**,**tlen**);**  **for** i**=**1**:**n  r**(**i**,:)=**L**(**i**)./(**1**+**exp**(-**k**(**i**)\***h**))-**L**(**i**)/**2**;**%生长率  **end**  figure  plot**(**h**,**r**),**xlabel**(**'降水量'**),**ylabel**(**'生长率'**),**  legend**(**'specie1'**,**'specie2'**,**'specie3'**,**'specie4'**,**'Location'**,**'best'**)**  beautiplot |



以上2幅图是不同的参数设置情况下的生长率随降水量变化的曲线，其中降水量已归一到0-1区间，大家可以自行调整参数绘图

### 微分方程参数设置

**下面是微分方程代码求解思路：**

总结一下公式，需要设置的变量有：

T：求解时间段0-T，求解离散时间间隔dt，求解点数T/dt

h(t)：降水量h(t)随在时间段0-T内的变化，1\*(T/dt)的向量

n：物种数量n（n种物种），标量

:物种i的最大生长率, 对n种物种，是一个 1\*n的向量

: 物种i的逻辑斯谛增长率或曲线的陡度，和物种i对干旱的适应性有关系，对n种物种，是一个1\*n的向量

：物种k对物种i的竞争系数，对n种物种，是一个n\*n的矩阵

：物种i的最大容量，对n种物种，是一个 1\*n的向量

X0: n种物种的初始容量，是一个 1\*n的向量

用matlab函数描述此微分方程：

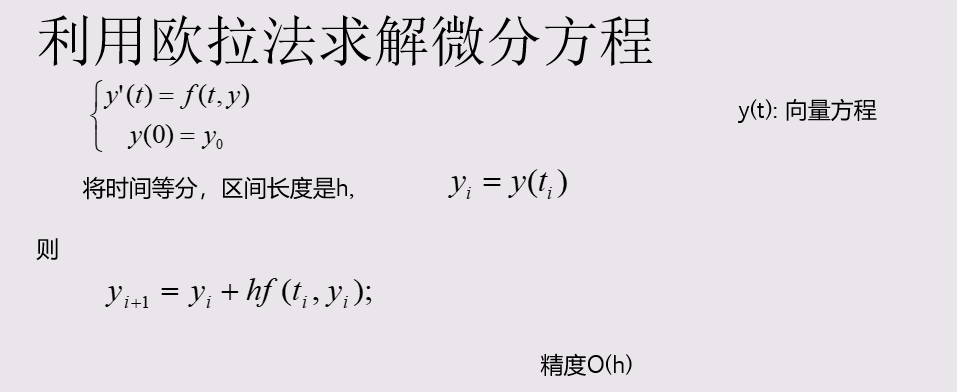
|  |
| --- |
| 问题1的微分方程模型 |
| **function** dx **=** problem1\_fun**(**t**,**x**,**n**,**h**,**L**,**k**,**alpha**,**N**)**  %问题1模型  dx **=** zeros**(**n**,**1**);**  **for** i**=**1**:**n  r\_i**=(**L**(**i**)/(**1**+**exp**(-**k**(**i**)\***h**(**t**)))-**L**(**i**)/**2**);**  tmp**=**0**;**  **for** kk**=**1**:**n  **if** kk**~=**i  tmp**=**tmp**+**alpha**(**i**,**kk**)\***x**(**kk**)/**N**(**kk**);**  **end**  **end**  tmp\_i**=**1**-**x**(**i**)/**N**(**i**)-**tmp**;**    dx**(**i**)=**r\_i**\***x**(**i**)\***tmp\_i**;**  **end**    **end** |

### 改进欧拉法

求解微分方程：改进欧拉法

微分方程的本质特征是方程中含有导数项，数值解法的第一步就是设法消除其导数值，这个过程称为离散化。 实现离散化的基本途径是用向前差商来近似代替导数，这就是欧拉算法实现的依据。

欧拉方法：



Step1

利用差商代替 得

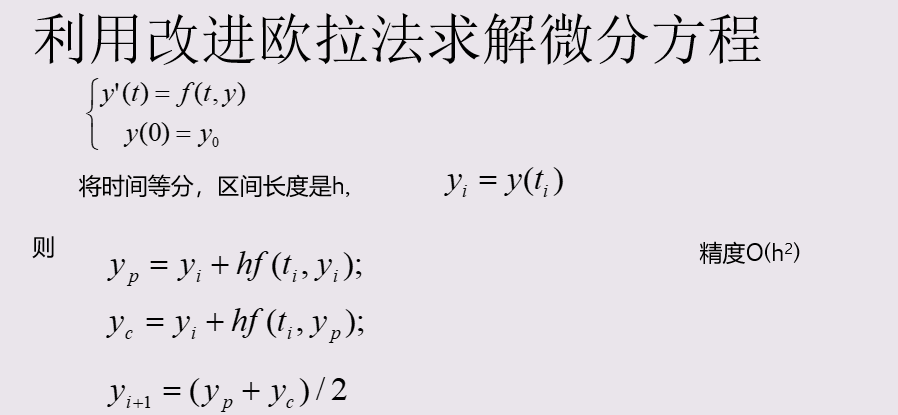


step2:

用 表示 的近似值，用 表示 的近似值，变为:



改进欧拉方法：



step1:

先用显式欧拉公式作预测，算出

step2:

再将代入隐式梯形公式的右边做校正得到

 为了方便编程计算，我们还可以将改进的欧拉公式写成：

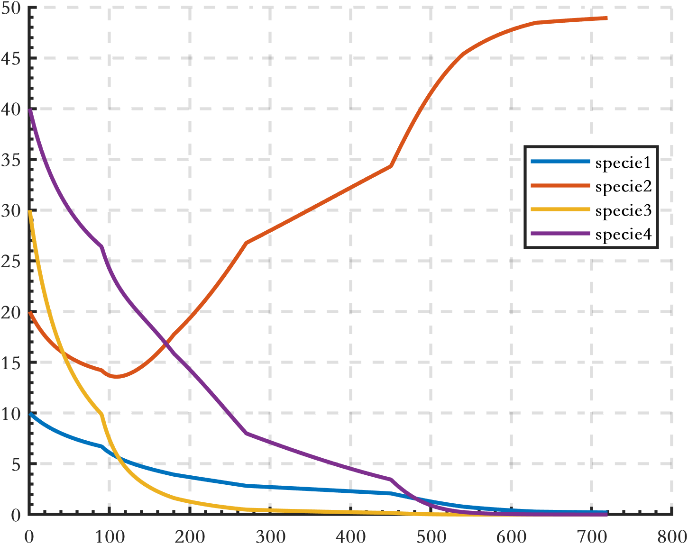
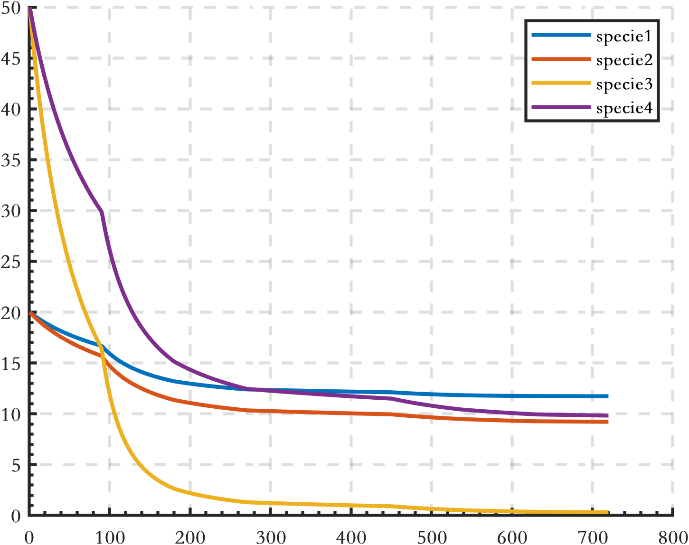


|  |
| --- |
| 改进欧拉法代码 |
| **function** **[**T**,**X**,**dX**]** **=** ODE\_ImprovedEuler**(** Hfun**,**t**,**h**,**x0 **)**  % [T,X,dX] = ODE\_ImprovedEuler( Hfun,t,h,x0 ) 改进欧拉法求解常微分方程  % Hfun为描述一阶微分方程导数的函数句柄，格式为 dX = Hfun( t,X )  % t为时间节点，可为标量，时间范围为 T = 0:h:t  % 长2向量 ，时间范围为 T = t(1):h:t(2)  % 向量 ，时间范围为 T = t  % h为时间步长，在t的前两种情况下，必须给出h具体值  % 直接给出时间节点t时，h可用[]或任意值占位  % x0为状态量初始值  % 算法：  % Xp = X(k-1) + h\*dX(k-1)  % dXp = Hfun( T(k),Xp )  % X(k) = X(k-1) + (h/2)\*[dX(k-1)+dXp]  **if** nargin **<** 4  error**(**'初始值必须给出'**);**  **end**  % 确定时间节点  n **=** length**(**t**);**  **if** n **==** 1  T **=** 0**:**h**:**t**;**  **elseif** n **==** 2  T **=** t**(**1**):**h**:**t**(**2**);**  **else**  T **=** t**;**  **end**  T **=** T**(:);** % 时间变为列向量  % 计算  N **=** length**(**T**);**  x0 **=** x0**(:);** x0 **=** x0**';** % 初值变为行向量  m **=** length**(**x0**);** % 状态量维数  X **=** zeros**(**N**,**m**);** % 初始化状态量  dX **=** zeros**(**N**,**m**);** % 状态导数  X**(**1**,:)** **=** x0**;**  **for** k **=** 2**:**N  dX**(**k**-**1**,:)** **=** Hfun**(** T**(**k**-**1**),**X**(**k**-**1**,:)** **);**  h **=** T**(**k**)** **-** T**(**k**-**1**);**  Xp **=** X**(**k**-**1**,:)** **+** h**\***dX**(**k**-**1**,:);**  dXp **=** Hfun**(** T**(**k**),**Xp **);**  X**(**k**,:)** **=** X**(**k**-**1**,:)** **+** **(**h**/**2**)\*(**dX**(**k**-**1**,:)+**dXp**');**  **end**  dX**(**N**,:)** **=** Hfun**(** T**(**N**),**X**(**N**,:)** **);**  **if** nargout **==** 0  plot**(**T**,**X**)**  **end** |

主函数代码：

|  |
| --- |
| 主函数代码（请勿直接复制运行，请运行problem1.m文件） |
| clc**;** clear**;** close all**;**  year**=**2**;**%模拟年数  T**=**360**\***year**;**%方便起见，一年设置为360天，一个季度90天  dt**=**1**;**%时间间隔：1天  tlen**=**T**/**dt**;**  tspan **=** linspace**(**1**,** T**,** tlen**);**  n**=**4**;**%物种数量  %%  L**=[**0.3**,**0.2**,**0.2**,**0.25**];**%物种i的最大生长率  k**=[**1**,**2**,**10**,**4**];**%物种i的逻辑斯谛增长率  h**=(**1**/**tlen**)\*(**1**:**tlen**);**%归一到0-1的降水量h(t)  r**=**zeros**(**n**,**tlen**);**  **for** i**=**1**:**n  r**(**i**,:)=**L**(**i**)./(**1**+**exp**(-**k**(**i**)\***h**))-**L**(**i**)/**2**;**%生长率  **end**  figure  plot**(**h**,**r**),**xlabel**(**'降水量'**),**ylabel**(**'生长率'**),**  legend**(**'specie1'**,**'specie2'**,**'specie3'**,**'specie4'**,**'Location'**,**'best'**)**  beautiplot  exportgraphics**(**gcf**,**'img/多物种情况下生长率关于降水量的函数图像1.png'**,**'Resolution'**,**300**)**  h**=[**0.01**\***ones**(**1**,**90**),**0.04**\***ones**(**1**,**90**),**0.03**\***ones**(**1**,**90**),**0.01**\***ones**(**1**,**90**)];**%设置一年内4个季度的降水量  h**=**repmat**(**h**,**1**,**year**);**%重复year年  alpha**=[**1**,**2**,**2**,**2**;**  2**,**1**,**2**,**2**;**  2**,**2**,**1**,**2**;**  2**,**2**,**2**,**1**];**%物种k对物种i的竞争系数  N**=[**50**,**50**,**50**,**50**];**%物种i的最大容量  x0**=[**20**,**20**,**50**,**50**];**%物种i的初始容量  % Solving the differential system of equations with ode45  % [t,x] = ode45(@(t,x) problem1\_fun(t,x,n,h,L,k,alpha,N), tspan, x0);  %改进欧拉法求解微分方程  **[**t**,**x**]** **=** ODE\_ImprovedEuler**(** **@(**t**,**x**)** problem1\_fun**(**t**,**x**,**n**,**h**,**L**,**k**,**alpha**,**N**),**tspan**,**dt**,**x0 **);**  figure  plot**(**t**,**x**);**xlabel**(**'时间/天'**),**ylabel**(**'植物数量'**)**  legend**(**'specie1'**,**'specie2'**,**'specie3'**,**'specie4'**,**'Location'**,**'best'**),**beautiplot  exportgraphics**(**gcf**,**'img/4种物种的数量生长曲线1.png'**,**'Resolution'**,**300**)**  L**=[**1.6**,**1.4**,**1.0**,**0.5**];**  k**=[**1**,**2**,**5**,**4**];**  h**=(**1**/**tlen**)\*(**1**:**tlen**);**  r**=**zeros**(**n**,**tlen**);**  **for** i**=**1**:**n  r**(**i**,:)=**L**(**i**)./(**1**+**exp**(-**k**(**i**)\***h**))-**L**(**i**)/**2**;**  **end**  figure  plot**(**h**,**r**),**xlabel**(**'降水量'**),**ylabel**(**'生长率'**),**  legend**(**'specie1'**,**'specie2'**,**'specie3'**,**'specie4'**,**'Location'**,**'best'**)**  beautiplot  exportgraphics**(**gcf**,**'img/多物种情况下生长率关于降水量的函数图像2.png'**,**'Resolution'**,**300**)**  h**=[**0.01**\***ones**(**1**,**90**),**0.04**\***ones**(**1**,**90**),**0.03**\***ones**(**1**,**90**),**0.01**\***ones**(**1**,**90**)];**  h**=**repmat**(**h**,**1**,**year**);**  alpha**=[**1**,**2**,**3**,**4**;**  4**,**1**,**3**,**2**;**  3**,**2**,**1**,**4**;**  2**,**3**,**4**,**1**];**  N**=[**100**,**50**,**120**,**130**];**  x0**=[**10**,**20**,**30**,**40**];**  **[**t**,**x**]** **=** ODE\_ImprovedEuler**(** **@(**t**,**x**)** problem1\_fun**(**t**,**x**,**n**,**h**,**L**,**k**,**alpha**,**N**),**tspan**,**dt**,**x0 **);**  figure  plot**(**t**,**x**);**xlabel**(**'时间/天'**),**ylabel**(**'植物数量'**)**  legend**(**'specie1'**,**'specie2'**,**'specie3'**,**'specie4'**,**'Location'**,**'best'**),**beautiplot  exportgraphics**(**gcf**,**'img/4种物种的数量生长曲线2.png'**,**'Resolution'**,**300**)** |

### 求解结果



以上是不同参数设置下的植物数量随时间的变化曲线情况，大家可以设置不同的参数进行运行

# 问题2与问题3：物种数量最优控制模型

问题2：就植物群落与大环境的长期相互作用，探讨你能从你的模型中得出什么结论。请考虑以下问题。 • 社区需要多少种不同的植物物种才能受益，随着物种数量的增加会发生什么？

社区需要多少种不同的植物物种才能受益，这明显是个最优化问题，建立最优物种竞争微分方程模型，

思考优化变量是什么：明显是植物物种的数量

思考目标函数是什么：长时间稳定状态下植物生物量的总和最大

什么是生物量，简单来说就是有机物的含量，有机物越多，对人就越有利；

但是问题1中没有和生物量有关的建模，怎么办？看第3问

这题可以和问题3一起建模和回答（建模时没有必要把这几个问题区分的这么明显，这点美赛和国赛不同，你要是把每个问题都详细建模和解释，那根本没有时间和精力）

问题3：社区中的物种类型如何影响你的结果？

## 问题分析

物种类型：暂定为草本、灌木、树木这三种，草本、灌木、树木的生物量明显是逐渐增加的，而种群数量（问题1中的）是逐渐减小的，我们可以收集数据，确定一棵草本、灌木、树木的生物量，比如一棵草的生物量是1，灌木为10，树木为100，对应物种类型的生物量乘以第一问的种群数量（问题1中的）就得到了总的生物量

## 模型建立

### 决策变量

草本、灌木、树木分别有多少种

### 目标函数

长时间稳定状态下植物生物量的总和尽可能大



其中b1，b2，b3分别是一棵草本、灌木、树木的生物量，为对应为草本的植物种类数，为对应为灌木的植物种类数，为对应为树木的植物种类数，所以总的植物种类数为

### 控制方程



### 约束条件

由于必须设置总的植物种类的上限，假设为（可以自己设置），不然无法求解，

所以约束条件1：

约束条件2：都为小于正整数

### 初始条件

每一种植物都有一个初始值

### 总体模型

**总的模型公式为：**



其中，目标函数中的表示只是求解实际最优的近似值。Argmax表示求最优参数，下标表示要求解的优化变量。

## 模型求解

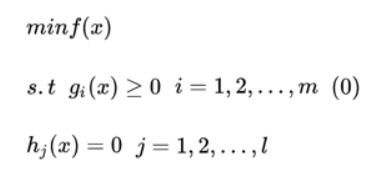
由于上面涉及到微分方程和约束条件，有微分方程代表这个模型是非线性的，需要使用遗传算法、粒子群、模拟退火等非线性的优化求解算法

### 处理约束条件：罚函数法

罚函数法（英语：penalty method）是求解有约束的最优化问题的一种算法。

罚函数法的要旨是将一个有约束的最优化问题转化为一系列的无约束问题；这些无约束问题由原问题及罚函数，再加上惩罚因子组成；而且，这些无约束问题的解会收敛于所求问题的解。

**优化问题的标准型如下图，是求最小值，而我们的模型是求最大值**



将目标函数变成标准型，即变为负数，再利用罚函数法，当时，目标函数加上一个罚因子，其中罚因子是一个很大的正数，即不满足约束时，目标函数将会变得很大，而我们是求最小值，因此不满足约束的情况会被剔除。而时，加上0



写作时，你可以把模型公式写到一块，就是公式那样，不过这是国赛的推荐写法，美赛我的推荐写法是分点明确写出最优问题的5大条件：

决策变量

目标函数

控制方程

约束条件

初始条件

前文的分析写得比较简略，大家可以把这5点详细展开，每一点都作为一个小标题，并加粗。我提供了去年我A题论文的中文版本，大家可以参考一下

**下面开始求解：**

**目录下我提供了最优化问题相关算法的代码**

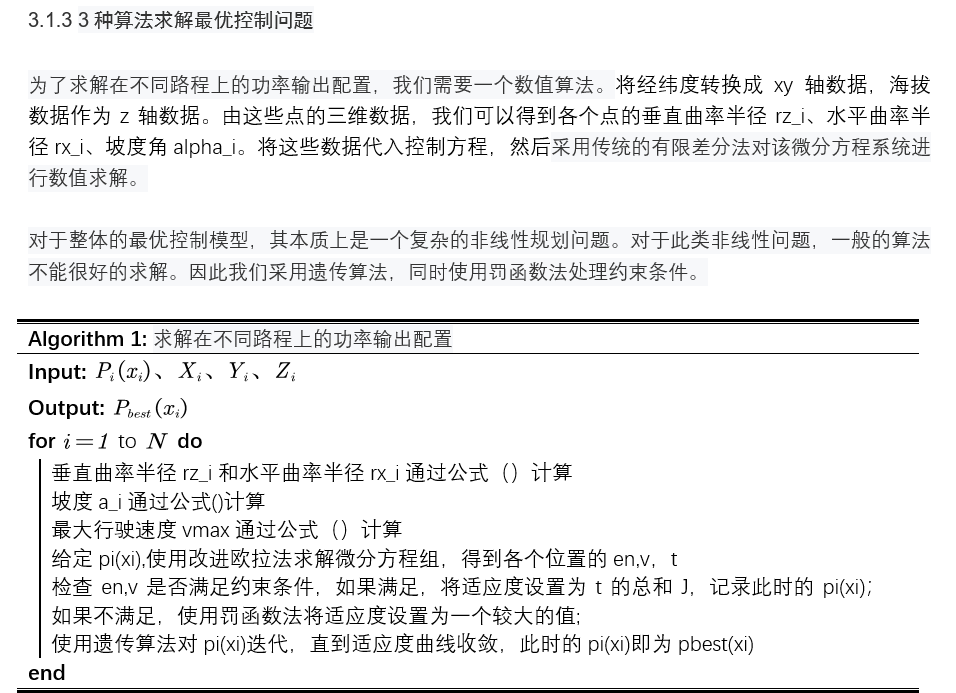
|  |
| --- |
| **7.高级优化算法**  **1）粒子群优化算法(求解无约束优化问题)**  **1>PSO(基本粒子群算法)**  **2>YSPSO(待压缩因子的粒子群算法)**  **3>LinWPSO(线性递减权重粒子群优化算法)**  **4>SAPSO(自适应权重粒子群优化算法)**  **5>RandWSPO(随机权重粒子群优化算法)**  **6>LnCPSO(同步变化的学习因子)**  **7>AsyLnCPSO(异步变化的学习因子)(算法还有bug)**  **8>SecPSO(用二阶粒子群优化算法求解无约束优化问题)**  **9>SecVibratPSO(用二阶振荡粒子群优化算法求解五约束优化问题)**  **10>CLSPSO(用混沌群粒子优化算法求解无约束优化问题)**  **11>SelPSO(基于选择的粒子群优化算法)**  **12>BreedPSO(基于交叉遗传的粒子群优化算法)**  **13>SimuAPSO(基于模拟退火的粒子群优化算法)**  **2)遗传算法**  **1>myGA(基本遗传算法解决一维约束规划问题)**  **2>SBOGA(顺序选择遗传算法求解一维无约束优化问题)**  **3>NormFitGA(动态线性标定适应值的遗传算法求解一维无约束优化问题)**  **4>GMGA(大变异遗传算法求解一维无约束优化问题)**  **5>AdapGA(自适应遗传算法求解一维无约束优化问题)**  **6>DblGEGA(双切点遗传算法求解一维无约束优化问题)**  **7>MMAdapGA(多变异位自适应遗传算法求解一维无约束优化问题)** |

**注：上面建立的模型不用想，肯定求不出来，去年美赛A题我也是建立了一个复杂的优化模型，花了很长时间但是解不出来，那还是已经有实际数据的情况下。而现在我们没有实际数据，第一问参数都是编的，那这里我们也可以直接编结果**

**就是在论文里面写我们实际使用了3个算法求解这个模型：**

1. **改进欧拉法求解微分方程**
2. **罚函数法处理约束条件，将有约束优化问题转变为无约束优化问题**
3. **遗传算法/粒子群/模拟退火/蚁群（以及上面各种改进的遗传算法、粒子群算法，推荐使用上面的改进算法，比较好水论文）等等算法求解无约束优化问题**

**看我去年是怎么写的：**



对于整体的最优问题模型，其本质上是一个复杂的非线性规划问题。对于此类非线性问题，一般的算法不能很好的求解。因此我们采用遗传算法（你可以改成粒子群、模拟退火等等），同时使用罚函数法处理约束条件。

然后我们可以写一下遗传算法、粒子群、模拟退火或者上面各种改进的智能算法的原理，最好使用伪代码

|  |
| --- |
| **Algorithm 1:** 遗传算法步骤 |
| **Input: 目标函数fitness、优化变量范围** |
| **Output: 最优优化变量及对应的目标函数值**  **for**  to  **do**   |  | | --- | | 补充你自己的算法 |   **end** |

### 遗传算法

遗传算法思想

借鉴生物进化论，遗传算法将要解决的问题模拟成一个生物进化的过程，通过复制、交叉、突变等操作产生下一代的解，并逐步淘汰掉适应度函数值低的解，增加适应度函数值高的解。这样进化N代后就很有可能会进化出适应度函数值很高的个体。

举个例子，使用遗传算法解决“0-1背包问题”的思路：0-1背包的解可以编码为一串0-1字符串（0：不取，1：取） ；首先，随机产生M个0-1字符串，然后评价这些0-1字符串作为0-1背包问题的解的优劣；然后，随机选择一些字符串通过交叉、突变等操作产生下一代的M个字符串，而且较优的解被选中的概率要比较高。这样经过G代的进化后就可能会产生出0-1背包问题的一个“近似最优解”。

编码：需要将问题的解编码成字符串的形式才能使用遗传算法。最简单的一种编码方式是二进制编码，即将问题的解编码成二进制位数组的形式。例如，问题的解是整数，那么可以将其编码成二进制位数组的形式。将0-1字符串作为0-1背包问题的解就属于二进制编码。

遗传算法有3个最基本的操作：选择，交叉，变异。

选择：选择一些染色体来产生下一代。一种常用的选择策略是 “比例选择”，也就是个体被选中的概率与其适应度函数值成正比。假设群体的个体总数是M，那么那么一个体Xi被选中的概率为f(Xi)/( f(X1) + f(X2) + …….. + f(Xn) ) 。比例选择实现算法就是所谓的“轮盘赌算法”( Roulette Wheel Selection ) ，轮盘赌算法的一个简单的实现如下：

**交叉(Crossover)**：2条染色体交换部分基因，来构造下一代的2条新的染色体。例如：

交叉前：

00000|011100000000|10000

11100|000001111110|00101

交叉后：

00000|000001111110|10000

11100|011100000000|00101

染色体交叉是以一定的概率发生的，这个概率记为Pc 。

**变异(Mutation)**：在繁殖过程，新产生的染色体中的基因会以一定的概率出错，称为变异。变异发生的概率记为Pm 。例如：

变异前：

000001110000000010000

变异后：

000001110000100010000

**适应度函数 ( Fitness Function )**：用于评价某个染色体的适应度，用f(x)表示。有时需要区分染色体的适应度函数与问题的目标函数。例如：0-1背包问题的目标函数是所取得物品价值，但将物品价值作为染色体的适应度函数可能并不一定适合。适应度函数与目标函数是正相关的，可对目标函数作一些变形来得到适应度函数。

### **遗传算法的伪代码**

基本遗传算法伪代码

/\*  
\* Pc：交叉发生的概率  
\* Pm：变异发生的概率  
\* M：种群规模  
\* G：终止进化的代数  
\* Tf：进化产生的任何一个个体的适应度函数超过Tf，则可以终止进化过程  
\*/  
初始化Pm，Pc，M，G，Tf等参数。随机产生第一代种群Pop  
  
do  
{  
　　计算种群Pop中每一个体的适应度F(i)。  
　　初始化空种群newPop  
　　do  
　　{  
　　　　根据适应度以比例选择算法从种群Pop中选出2个个体  
　　　　if ( random ( 0 , 1 ) < Pc )  
　　　　{  
　　　　　　对2个个体按交叉概率Pc执行交叉操作  
　　　　}  
　　　　if ( random ( 0 , 1 ) < Pm )  
　　　　{  
　　　　　　对2个个体按变异概率Pm执行变异操作  
　　　　}  
将2个新个体加入种群newPop中  
} until ( M个子代被创建 )  
用newPop取代Pop  
}until ( 任何染色体得分超过Tf， 或繁殖代数超过G )

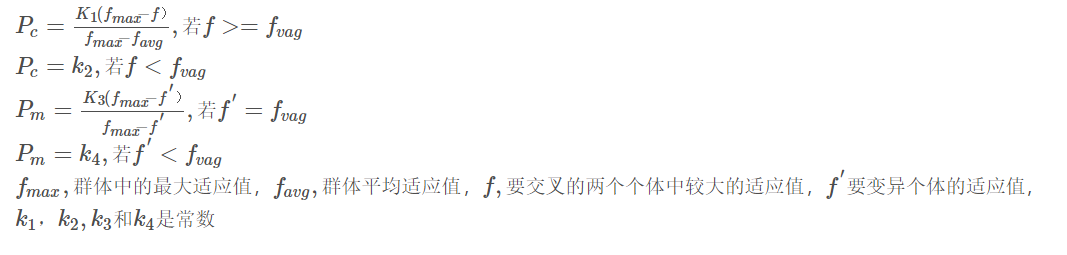
### 多变异位自适应遗传算法

你也可以写其他优化算法，比如第二三问代码压缩包的最优化问题相关算法目录下有多变异位自适应遗传算法，参考：https://blog.csdn.net/xuehuafeiwu123/article/details/52332723

算法原理

自适应遗传算法是交叉概率和变异概率能够随使用度自动改变，以求得相对某个解的最佳交叉概率和变异概率。本算法是在自适应遗传算法中引进多变异位，以增加种群的多样性。

**自适应遗传算法**中的交叉概率和变异概率的计算公式为（记得写论文时把公式写成伪代码的形式）：



多变异位是指变异位的二进制表示的编码的多个位取反。

算法步骤

1、随机产生种群，

2、用轮盘赌策略确定个体的适应度，判断是否符合优化准则，若符合，输出最佳个体及其最优解，结束，否则，进行下一步

3、依据适应度选择再生个体，适应度高的个体被选中的概率高，适应度低的个体被淘汰

4、按照一定的交叉概率和交叉方法，生成新的个体

5、通过自适应方法产生变异概率，若最大适应率等于最小适应率，则只产生一个变异位，否则，随机产生变异位的个数和位置，对选中的个体进行变异

6、由交叉和变异产生新一代种群，返回步骤2

### 整体的优化问题求解

对于整体的优化问题求解，我们同样可以写一段伪代码：

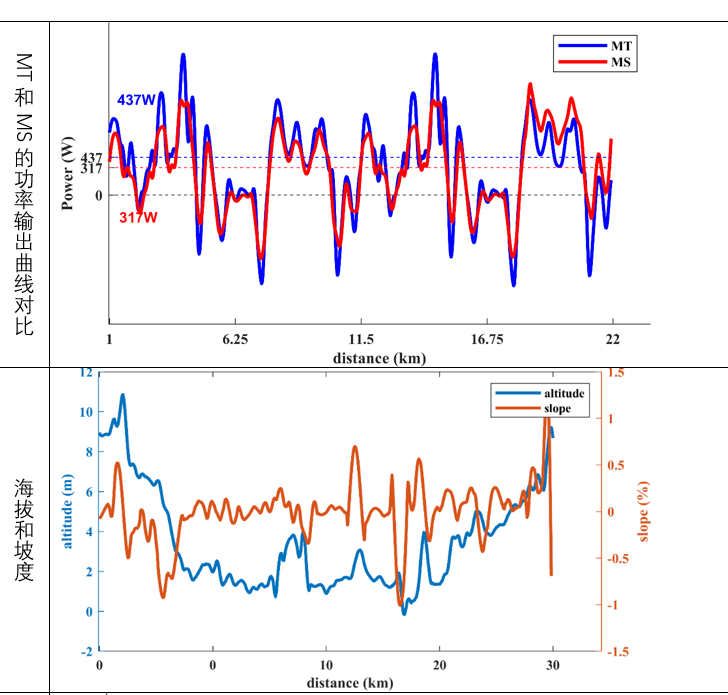
|  |
| --- |
| **Algorithm 1:** 求解草本、灌木、树木的物种数量 |
| **Input:** |
| **Output:**  **for**  to  **do**   |  | | --- | | ….使用改进欧拉法求解微分方程组，得到各种植物随时间的变化曲线  检查……是否满足约束条件，如果满足，将适应度fitness设置为实际的目标函数，如果不满足，使用罚函数法将适应度加上,使其变为一个较大的值;  使用遗传算法对……迭代，直到适应度曲线收敛，此时的pi(xi)即为pbest(xi) |   **end** |

## 求解结果

**然后贴上结果图：**

**求解结果**

**因为我们没有实际求解结果，所以直接编结果，比如是去年我们的结果就是随机生成的：**



**曲线怎么编，先用常识想一下结果大概是什么样，然后随机生成这样的曲线；或者使用表格表达最后的求解结果，表格数据更好编一点。**

**举个编数据的例子：**

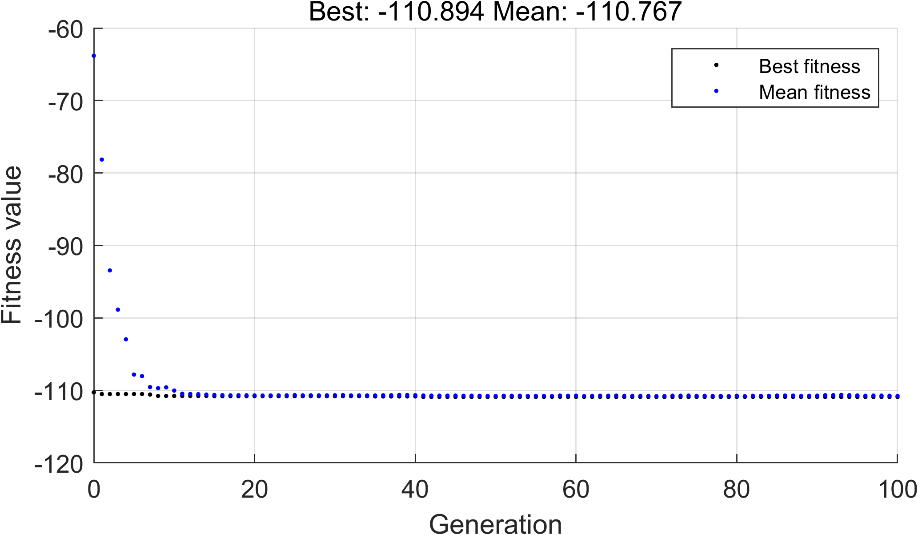
**我们使用遗传算法/粒子群算法对模型求解多次：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实验次数** | **最优草本物种数** | **最优灌木物种数** | **最优树木物种数** | **最大生物量** |
| **1** | **10** | **7** | **4** | **18753** |
| **2** | **8** | **8** | **5** | **19357** |
| **3** | **12** | **10** | **6** | **20567** |
|  |  |  |  |  |
|  |  |  |  |  |

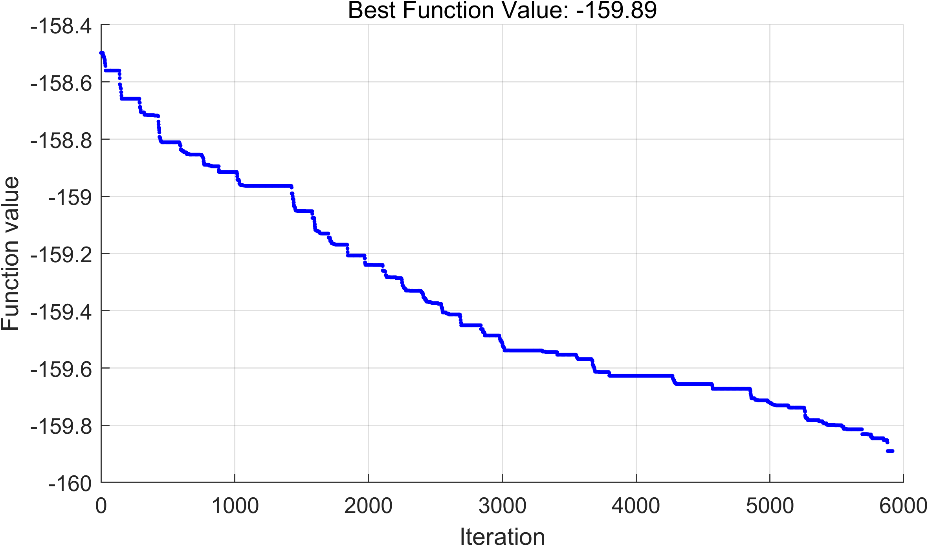
**其中第三次模拟的算法迭代图如下：**

**算法迭代图：遗传算法/粒子群算法的迭代图，类似下面这种**

**遗传算法：**



**模拟退火算法：**



由于第三次实验得到的最大生物量最大，为20567（单位），因此我们选择第三次实验的结果，即当草本、灌木、树木物种数量分别为12、10、6时，社区能够受益

# 问题4

在未来的天气周期中，干旱发生的频率更高、变化更大会产生什么影响?如果干旱不那么频繁，那么物种数量对整体种群的影响是否相同?

翻译成人话：

频率更高：一段时间内设置降水量较少的时间段的数量

变化更大：降水量较少的时间段的持续时间，降水量减少的程度

对第一问设置不同的降水量情况，求解不同降水量情况（降水量较少的时间段的数量和降水量减少的程度不同）的种群数量变化

# 问题5

污染和栖息地减少等其他因素如何影响你的结论？

2个解决方法：

将污染和栖息地减少等其他因素综合进问题1的模型，设置相关的参数进行模型

把这一节写到最后，语文建模

注意，对于这一问，我们只需要写出公式即可，简单分析，不需要像问题2和问题3一样详细分析

# 问题6

你的模型表明，为确保一个植物群落的长期生存能力，应该做些什么，对大环境有什么影响？

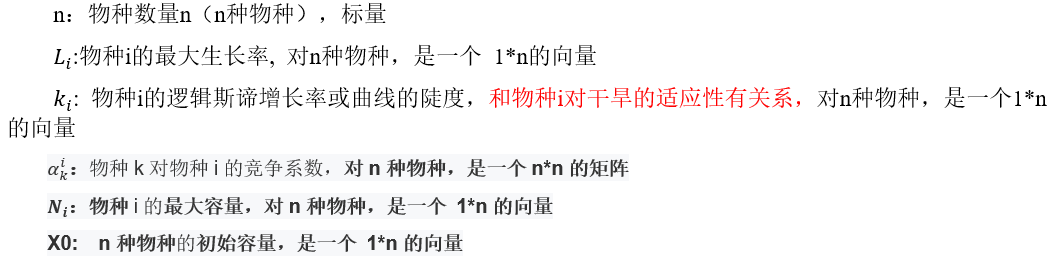
根据前面几个问题的结论进行适当的语文建模

# 灵敏性分析

最重要的是灵敏性分析，没写灵敏性分析基本就是S奖了

但幸运的是，灵敏性分析最好编了

在问题1中有很多参数，比如Li、ki、、Ni



这几个变量是我们设置的超参数，分析这几个变量对问题2和问题3中最优解（草木、灌木、树木的物种数量）的影响

将这几个变量减少和增加1%，2%，3%，4%，5%，草木、灌木、树木的最优物种数量会如何变化的曲线图

套话如下：

在我们的模型中，……是影响结果的因素之一，……

另外，……也可能对我们的模型结果产生影响，因此我们对……进行灵敏性分析。

通过在这3个因素标准值上下改变5%，并且相同的改变做多次模拟实验，求目标函数的均值和误差，结果如图：

去年我的灵敏性分析图，就是画了一个误差图，数据都是编的

