Projeto 1: Administração Florestal

Gabriel Mota Lima
11794870

Herberth Luan Vieira Oliveira
12559110

Mateus Queiroz de Souza Daniel
11294552

Victor Viana de Oliveira Matos
11810821

Vinícius da Costa Collaço
11811012

4 de junho de 2022

1 Introdução

A sustentabilidade nunca foi tão discutida como é atualmente, são reuniões com líderes políticos e leis criadas para que cada vez se tenham a conscientização da utilização dos recursos dados da natureza de maneira mais responsável possível, sendo que são finitos.

Com isso em mente o uso de modelos matemáticos juntamente com o poder computacional ficaram mais procurados, visto que uma parte da matemática trabalha com resolução de problemas de máximos e mínimos e deste modo auxiliando diversas áreas para que façam os seus processos de maneira a não desperdiçar materiais que são utilizados nos processos, ou seja, otimização do uso.

Nesse trabalho é apresentado modelo matemático dado um problema de plantio de árvores, o qual objetivo será o modelo deixar o teu plantio sustentável e otimizando o retorno para plantador.

1.1 Situação-problema e conceitos

O contexto que será aplicado modelo matemático é de uma área, onde encontra-se um vetor inicial x de formação de árvores de variadas alturas n, as quais terão um crescimento em determinado período. Após o crescimento será feita a colheita e replantio delas. O problema consiste em realizar um modelo matemático que tenha um corte sustentável e que economicamente se tenha um retorno sustentável ótimo para o proprietário.

Alguns conceitos se fazem importante explicitá-los para que o modelo matemático fique mais entendível, seriam: corte sustentável e retorno sustentável ótimo.

Corte sustentável, conforme Anton Rorres, pode-se definir o corte sustentável quando após o processo de crescimento e colheita a configuração final seja igual a tua inicial, ou seja, retornar ao vetor inicial x. A Figura 1 abaixo mostra de maneira mais lúcida o termo de corte sustentável:

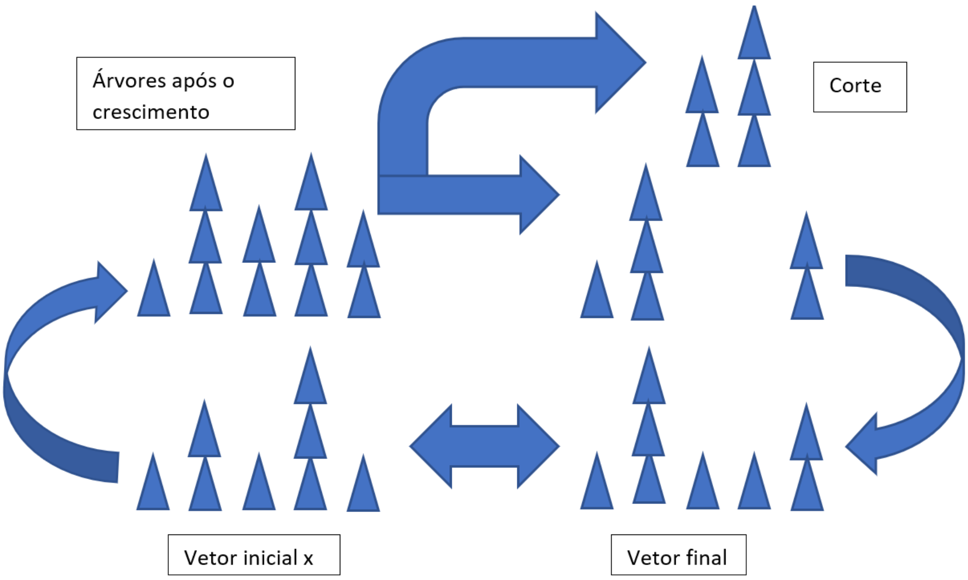


Figura 1: Colheita Sustentável

Retorno Sustentável Ótimo é definido por um teorema, de acordo com Anton Torres, o rendimento é obtido cortando todas as árvores de uma classe de altura específica e nenhuma árvore de qualquer outra classe. Portanto, esse conceito será a base para o modelo matemático consiga otimizar o corte economicamente para o plantador.

1.2 Modelo Matemático

Por definição, modelo é uma representação da realidade que acaba sendo simplificada e idealizada, portanto, algumas restrições que serão colocadas para que seja aplicável e atendido a situação-problema.

O primeiro momento a organização das árvores é a partir de classes avaliando pela altura, sendo que a primeira classe será a muda a qual foi passado que não há custos e nem receita para essa classe e que dada classe n não há mais crescimento. Os valores econômicos são função das respectivas alturas.

Dentro do modelo há algumas restrições que seguem:

• cada árvore no teu período de crescimento só consegue crescer de uma classe, exceto àquelas que estejam na classe n, pois essa se mantém;

• nenhum uma árvore morre nenhum dos processos;

• os preços são fixados;

• as taxas de crescimento são fixadas para cada classe

2 Equações que definem uma colheita sustentável

Para termos uma colheita sustentável, a configuração final da floresta(após o corte e plantio das mudas) deve ser igual à configuração inicial (floresta antes do crescimento). Como, no nosso modelo, o número de árvores é constante e igual a s, podemos descrever a configuração inicial da floresta como sendo um vetor coluna x, chamado de vetor de não cortadas. $\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ onde $\mathrm{x_i}$ representa a quantidade de árvores da classe i (sendo que as árvores são classificadas pelo intervalo de altura a que pertencem e a classe 1 corresponde a mudas sem valor comercial) e $\mathrm{s} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{x_i}$.

Após cada colheita, as árvores passam por um período de crescimento, onde uma fração $\mathrm{g_i}$ das árvores de cada classe i sobe para a classe i+1, obtendo-se assim a configuração pré-corte.

O número de árvores em cada classe i, na configuração pré-corte, é igual à soma entre o número de árvores da classe i-1 que cresceram $(\mathrm{g_{i-1}} \cdot \mathrm{x_{i-1}})$ e o número de árvores da classe i que não cresceram $((1 - \mathrm{g_i}) \cdot \mathrm{x_i})$, com exceção da classe 1, onde restarão apenas as mudas que não cresceram, e da classe n onde apenas serão somadas as árvores que cresceram da classe n-1. Assim obtemos o vetor da configuração pré-corte:

$$\mathrm{Z} = \begin{bmatrix} x_1 \cdot (1 - g_1) \\ x_2 \cdot (1 - g_2) + x_1 \cdot g_1 \\ \vdots \\ x_n + x_{n-1} \cdot g_{n-1} \end{bmatrix}$$

No entanto, convém escrever o vetor pré-corte como o o produto entre uma "matriz de crescimento"(G) e o vetor x, sendo que: $\mathrm{G} = \begin{bmatrix} 1-g_1 & 0 & 0 & & \cdots & 0 \\ g_1 & 1-g_2 & 0 & & \cdots & 0 \\ 0 & g_2 & 1-g_3 & & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-g_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & g_{n-1} & 1 \end{bmatrix}$

O vetor de árvores cortadas é dado por y. $\mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$ onde cada $\mathrm{y_i}$representa a quantidade de árvores cortadas na classe i.

Para que haja uma colheita sustentável, será plantado um número de mudas igual ao número de árvores retiradas.

Assim,sabemos que o vetor reposição a ser somado ao vetor pré-corte $(\mathrm{G} \cdot \mathrm{x})$ e ao vetor -y para se obter o vetor x deve possuir um valor igual à soma do número de árvores cortadas em todas as classes (exceto da classe 1, composta por mudas sem valor comercial) em sua primeira linha:

$$\mathrm{F} = \begin{bmatrix} \sum_{i=2}^{n} y_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Portanto podemos definir uma matriz de reposição R, de ordem n, como sendo a matriz que faz real a igualdade $F = R \cdot \mathrm{y}$. $\mathrm{R} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$

Com todas as variáveis do modelo podemos escrever matematicamente a equação que caracteriza uma política de corte sustentável.

$$\mathrm{Gx} - \mathrm{y} + \mathrm{Ry} = \mathrm{x}$$

ou também reescrita como

$$(\mathrm{I} - \mathrm{R})\mathrm{y} = (\mathrm{G} - \mathrm{I})\mathrm{x} \tag{1}$$

Colocando y em função de x tem-se: $\mathrm{y} = (\mathrm{I} - \mathrm{R})^{-1}(\mathrm{G} - \mathrm{I})\mathrm{x}$

onde G e R e a matriz identidade I possuem dimensões $n \times n$, x e y são matrizes coluna com dimensão $n \times 1$, sendo n o número de classes de árvores da floresta.

como $y_1 > 0$ representaria o corte de mudas, sem valor econômico e reposição por novas mudas, então deduzimos que:

$$y_1 = 0 \tag{2}$$

Com o resultado de (2), equação (1) pode ser escrita como um conjunto de equações da seguinte maneira:

$$\begin{array}{ccc} y_2 + y_3 + \cdots + y_n & = & g_1 x_1 \\ y_2 & = & g_1 x_1 - g_2 x_2 \\ y_3 & = & g_2 x_2 - g_3 x_3 \\ & \vdots & \\ y_{n-1} & = & g_{n-2} x_{n-2} - g_{n-1} x_{n-1} \\ y_n & = & g_{n-1} x_{n-1} \end{array} \tag{3}$$

nesse sistema temos que a primeira equação é a soma das demais, pois os valores de g_n, x_n e y_n são positivos, então as equações em (3) exigem que:

$$g_1x_1 \geq g_2x_2 \geq \cdots \geq g_{n-1}x_{n-1} \geq 0 \tag{4}$$

Portanto, para termos uma colheita sustentável, x e y devem satisfazer a equação (1) e as entradas do vetor coluna x devem satisfazer à inequação (4), para que a floresta tenha uma configuração tal qual permita um corte sustentável.

Com o resultado obtido podemos criar um algoritimo que dado as entradas: n (número de classes da floresta), X (vetor de configuração da floresta, tamanho n), G (vetor de crescimento da floresta, tamanho n-1) e Y (vetor colheita, tamanho n), ele nos retorna True se a colheita é sustentável, ou False se a colheita não é sustentável.

```
def EhSustentavel(n,X,G,Y):
  #Valores de entrada, n valor inteiro; X,G e Y, vetores
  import numpy as np

  #Tratando as entradas e criando a matriz identidade
  I = np.identity(n) #Matriz identidade n x n
  G_n_mais1 = G.copy()
  G_n_mais1.append(0) #vetor G com n+1 para gerar a matriz G n x n
  G = np.array(G)
  Gmc = np.atleast_2d(G).T #transforma o vetor G em matriz coluna
  X = np.array(X)
  Xmc = np.atleast_2d(X).T #transforma o vetor X em matriz coluna
  Y = np.array(Y)
  Ymc = np.atleast_2d(Y).T #transforma o vetor Y em matriz coluna

  #Matriz G n x n vinha do vertor G n-1
  GMatriz = I-(G_n_mais1*I)
  for i in range (n-1):
    GMatriz [i+1,i]=Gmc[i,0]
  GMatriz [n-1,n-1] = 1

  #Matriz R
  R = np.zeros((n,n))
  for i in range (n):
    R[0,i] = 1

  #Vetor GX para comparacao
  GX = []
  for i in range (n-2,-1,-1):
    GX.append(G[i]*X[i])
  GX = np.array (GX) #array com valores de g de n * x de n do ultimo
   para o primeiro
```

```
#Operacoes colheita sustentavel para comparacao posterior
valor1 =(I-R).dot(Ymc)
valor2 = (GMatriz-I).dot(Xmc)


#Condicoes Para Colheita sustentavel
if Y[0] != 0: return False #Se o primeiro valor de Y for 0 retorna False
elif GX[0] < 0: return False
elif np.all(np.diff(GX) > 0) == False: return False # retorna False se nao for g1x1>g2x2>...
elif (valor1.round()  == valor2.round()).all() == False: return False #Retorna falso se a floresta nao se manter igual apos a colheita
else: return True
```

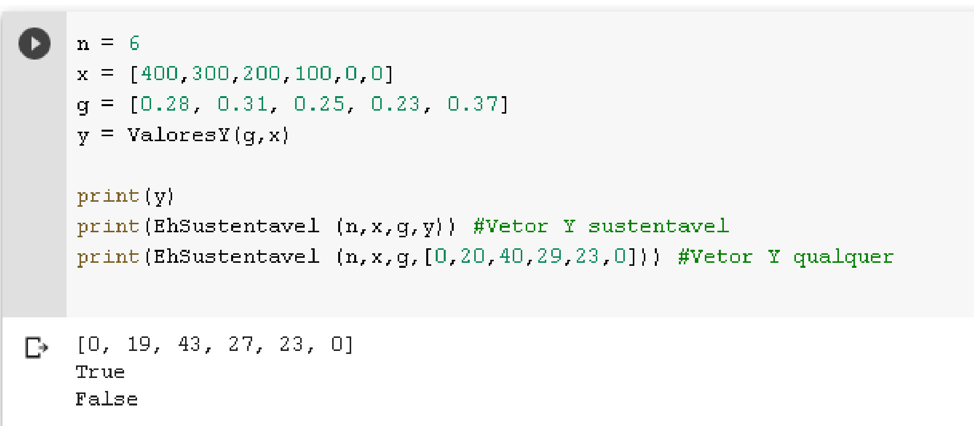
Listing 1: Funcão para definir se um corte é ou não sustentável

Com base em (2) e (3) podemos criar um algoritimo simples para retorna o vetor y, que é o vetor da colheita sustentável dado uma certa configuração x da floresta e um crescimento G:

```
def ValoresY (g,x):
  y = [0]#primeiro valor sempre 0
  n = len(x)

  for i in range (1,n-1):
    y.append((g[i-1]*x[i-1])-(g[i]*x[i]))
    y[i]=round(y[i])#arrendondar os valores
  y.append(g[-1]*x[-2])# yn = gn-1*xn-1
  y[-1]=round(y[-1])
  return y
```

Listing 2: Retorna o vetor Y



```
n = 6
x = [400,300,200,100,0,0]
g = [0.28, 0.31, 0.25, 0.23, 0.37]
y = ValoresY(g,x)

print(y)
print(EhSustentavel (n,x,g,y)) #Vetor Y sustentavel
print(EhSustentavel (n,x,g,[0,20,40,29,23,0])) #Vetor Y qualquer
```

```
[0, 19, 43, 27, 23, 0]
True
False
```

Figura 2: Saida de Teste das funções anteriores

3 Modelando o problema em questão

3.1 Função objetivo

O principal objetivo do modelo aqui proposto é maximizar o retorno financeiro da colheita, dadas as restrições que propiciam uma colheita sustentável e outras restrições práticas.

Sendo o retorno de cada colheita um somatório dos produtos entre a quantidade de árvores cortadas de cada classe e seus respectivos preços, tem-se que o retorno financeiro total de cada colheita é dado por (lembrando que $y_1 = 0$):

$$RT \;=\; p_2 y_2 + p_3 y_3 + ... + p_n y_n$$

Então, usando-se as equações em (3) tem-se que a função objetivo do modelo proposto é dada por:

$$RT \;=\; p_2 g_1 x_1 + (p_3 - p_2)\, g_2 x_2 + ... + (p_n - p_{n-1})\, g_{n-1} x_{n-1} \tag{5}$$

A função RT é a função de **retorno sustentável** deve ser maximizada com relação às variaveis x_1, x_2, ... ,x_n para que se encontre o **retorno sustentável ótimo**.

3.2 Restrições

Uma das restrições que se pode estabelecer está relacionada com a capacidade total de plantio da floresta, isto é, a quantidade total de árvores plantadas não deve exceder a um determinado valor. No caso deste problema, considera-se que a quantidade de árvores no inicio do período de crescimento é sempre constante e igual a s, isto é:

$$x_1 + x_2 + ... + x_n = s \tag{6}$$

O conjunto de desigualdades apresentados em (4) são condições necessárias e suficientes para que a configuração estabelecidade pelo vetor x permita que ocorra uma colheita sustentável. Portanto, (4) representa mais um conjunto de restrições do nosso modelo.

O último conjunto de restrições está associado ao fato de que o número de árvores em cada classe no inicio do período de crescimento não pode ser negativo, isto é:

$$x_1,\, x_2,\, ...\, ,x_n \geq 0 \tag{7}$$

3.3 O modelo

Assim, dada a função objetivo (5) e as restrições (4), (6) e (7) apresentadas acima pode-se modelar o problema da seguinte forma:

$$\begin{cases} Max\ RT = p_2g_1x_1 + (p_3 - p_2)\,g_2x_2 + ... + (p_n - p_{n-1})\,g_{n-1}x_{n-1} \\ sujeito\ a: \\ x_1 + x_2 + ... + x_n = s \\ g_1x_1 \geq g_2x_2 \geq \cdots \geq g_{n-1}x_{n-1} \geq 0 \\ x_1,\ x_2,\ ...\ ,x_n \geq 0 \end{cases} \tag{8}$$

O modelo descrito em (8) é um problema de programação linear, pois tanto a função objetivo quanto as restrições são de grau 1 com respeito às variaveis $x_1,\ x_2,\ ...\ ,x_n$.

4 Encontrando o retorno sustentável ótimo

4.1 Teorema 1

Para encontrar o retorno ótimo será considerado o seguinte teorema:

"O rendimento sustentável ótimo é obtido cortando todas as ávores de uma classe de altura específica e nenhuma árvore de qualquer outra classe"(ANTON and RORRES, 2012).

4.2 Função de retorno ótimo como implicação do Teorema 1

Considerando que a classe k é a única que tem as árvores completamente cortadas, então todos os y_i $(i \neq k)$ devem ser iguais a zero, isto é:

$$y_2 = y_3 = ... = y_{k-1} = y_{k+1} = ... = y_n = 0 \tag{9}$$

Somente $y_k \neq 0$.

Estamos considerando que a cada período de crescimento as ávores de uma determinada classe só podem crescer para a classe imediatamente superior. Se sempre cortarmos todas as árvores da classe k, então nunca haverá árvores nas classes superiores a k. E após o corte, também não haverá nenhuma árvore na classe k. Então tem-se:

$$x_k = x_{k+1} = ... = x_n = 0 \tag{10}$$

Se substituirmos (9) e (10) na equação (3) chegamos ao seguinte resultado:

$$\begin{cases} y_k = g_1 x_1 \\ 0 = g_1 x_1 - g_2 x_2 \\ 0 = g_2 x_2 - g_3 x_3 \\ \vdots \\ 0 = g_{k-2} x_{k-2} - g_{k-1} x_{k-1} \\ y_k = g_{k-1} x_{k-1} \end{cases} \implies \begin{cases} y_k = g_1 x_1 \\ g_1 x_1 = g_2 x_2 \\ g_2 x_2 = g_3 x_3 \\ \vdots \\ g_{k-2} x_{k-2} = g_{k-1} x_{k-1} \\ g_{k-1} x_{k-1} = y_k \end{cases}$$

Isto é:

$$y_k = g_1 x_1 = g_2 x_2 = ... = g_{k-1} x_{k-1} \tag{11}$$

A partir do resultado (11) acima, colocando todos os x_i $(i \neq 1)$ em função de x_1, tem-se:

$$x_2 = \frac{g1 \cdot x_1}{g_2}, \; x_3 = \frac{g1 \cdot x_1}{g_3}, \, ... \, , x_{k-1} = \frac{g_1 \cdot x_1}{g_{k-1}}$$

Substituindo as expressões acima na equação (6), pode-se isolar o valor de x_1:

$$\begin{aligned} x_1 + \frac{g1 \cdot x_1}{g_2} + \frac{g1 \cdot x_1}{g_3} + ... \, + \frac{g_1 \cdot x_1}{g_{k-1}} &= s \\ x_1 \cdot \left(1 + \frac{g_1}{g_2} + \frac{g_1}{g_3} + ... + \frac{g_1}{g_{k-1}}\right) &= s \\ x_1 &= \frac{s}{1 + \frac{g_1}{g_2} + \frac{g_1}{g_3} + ... + \frac{g_1}{g_{k-1}}} \\ x_1 &= \frac{s}{g_1 \cdot \left(\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + ... + \frac{1}{g_{k-1}}\right)} \end{aligned} \tag{12}$$

Como somente serão cortadas as árvores da classe k, então o retorno ótimo será dado pelo preço das árvores da classe k (isto é p_k) multiplicado pela quantidade de árvores removidas (y_k). Então:

$$RT = p_k \cdot y_k$$

Pela relação (11), $y_k = g_1 x_1$, então:

$$RT = p_k \cdot g_1 \cdot x_1 \tag{13}$$

Substituindo (12) na relação (13) acima, tem-se:

$$RT = p_k \cdot g_1 \cdot \left[\frac{s}{g_1 \cdot \left(\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + ... + \frac{1}{g_{k-1}} \right)} \right]$$

Portanto, tem-se:

$$RT_k = \frac{p_k \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + ... + \frac{1}{g_{k-1}}} \tag{14}$$

Desta forma, testa-se a função (14) para todos as classes de árvores. A classe k que retornar o maior valor de RT_k é a que deve ser completamente remotida.

4.3 Valor do retorno ótimo

Escrevendo matematicamente o resultado encontrado anteriormente:

Definições:
n: quantidade de classes de altura para as árvores.
s: quantidade total de árvores no periodo de crescimento.
$\{p_2, p_3, ... , p_n\}$: preços das árvores em cada classe.
$\{g_1, g_2, ... , g_{n-1}\}$: parâmetros de crescimento das $n-1$ primeiras classes.

Dadas as definições acima, o retorno sustentável ótimo pode ser escrito matematicamente como:

$$Ret = Maximo \left\{ \frac{p_k \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + ... + \frac{1}{g_{k-1}}} \right\}, (k = 2, 3, ..., n) \tag{15}$$

4.4 Classe que será cortada

A classe que será cortada é a classe k $(k = 2, 3, ..., n)$ tal que RT_k é máximo.

4.5 Algoritmo em Python para encontrar o retorno ótimo

A relação demonstrada em (15) apresenta um algoritmo que demanda um processo de iteração. Pode-se fazer esta iteração em uma linguagem de programação. O código a seguir representa a definição da função *retorno_otimo* na linguagem Python (linguagem usada durante todo este trabalho), que tem como parâmetros g e p, que recebem como argumentos objetos Python do tipo *list*, contendo respectivamente os parametros de crescimento e os preços das ávores de cada classe (com exceção de g_n e p_1). Há também os parâmetros n e s, que recebem como argumentos, respectivamente, a quantidade de classes de alturas das árvores e a quantidade total de árvores no período de crescimento. A função itera por todos os valores de $k = 2, 3, ..., n$ e calcula todos os valores de RT_k.

```python
def retorno_otimo(n, s, p, g):
  retornos = []
  for i in range(n-1): #se tem 6 classes, i itera entre [0,1,2,3,4]
    soma_inversos_g = 0
    for j in range(i+1): # loop para calcular o denominador de RT_k
      soma_inversos_g = soma_inversos_g + 1/g[j]
    r = p[i]*s / soma_inversos_g # calcula RT_k para cada classe k
    retornos.append(round(r, 2)) # armazena os retornos para cada classe
  return (retornos, max(retornos), 2+retornos.index(max(retornos)))
```

Listing 3: Função retorno_otimo

A função acima retorna três objetos:

retornos: lista contendo os todos os valores de RT_k pra todos os valores de k possíveis.

max(retornos): função que retorna o valor numérico do maior retorno, isto é, retorna o maior valor de RT_k.

2+retornos.index(max(retornos)): índice da classe que será cortada. Soma-se 2 ao índice do maior retorno, pois a iteração é feita com base na lista de preços que começa em p_2, isto é, quando $k = 2$ o indice do vetor retornos é 0.

Assim, por meio da função acima, o retorno sustentável ótimo é conseguido através da segunda posição da *tupla* retornada pela chamada da função. Por exemplo:

```python
_, Ret, classe_cortada = retorno_otimo(classes, s, precos, g)
```

Listing 4: Encontrando o retorno ótimo

A variável *Ret* por ser a segunda posição na atribuição das variáveis irá armazenar o retorno ótimo. E por seguinte, a variável *classe_ cortada* armazenará o número correspondente a classe que será cortada. A chamada de função demonstrada na *Listing 4*, bem como a função escrita na *Listing 3* serão usados para resolver o exercício prático seguinte.

5 Resolução de um problema prático

É dado um problema com os seguintes valores para os parâmetros g, p e n:

$$g_1 = 0.28, g_2 = 0.31, g_3 = 0.25, g_4 = 0.23 e g_5 = 0.37$$

$$p_2 = 50, p_3 = 100, p_4 = 150, p_5 = 200, p_6 = 250$$

$$n = 6$$

Dados estes valores, falta apenas o valor de s para que possamos substituir na função da *Listing 3* que retrata o algoritmo (15) e encontrar o retorno ótimo. Como todos os valores de RT_k estão em função do parâmetro s, para que se encontre o maior valor de

RT_k não importa o valor de s. Então escolhemos um valor arbitrário para substituir na relação e testar no algoritmo em Python, neste caso $s = 1000$.

5.1 Resolução em Python usando o resultado do Teorema 1

Foram criadas variáveis para armazenar os valores referidos acima, são elas:

```
g = [0.28, 0.31, 0.25, 0.23, 0.37]
precos = [50, 100, 150, 200, 250]
classes = 6
s = 1000 #valor arbitrario
```

Listing 5: Definindo os valores das variáveis

Em seguida chamamos a função *retorno_otimo* definina em *Listing 3*, e passamos a ela os argumentos definidos em *Listing 5*. Armazena-se então os valores que esta função retorna nas seguintes variáveis:

```
vetor_retornos, Ret, classe_cortada = retorno_otimo(classes,s,precos,g)
```

Listing 6: Chamando a função *retorno_otimo* e armazenando retornos

5.1.1 Classe cortada

Como retorno da função, será armazenado na variável *classe_cortada* o número correspondente a classe que será cortada completamente.

Observando pelo console do Python, tem-se que o valor da classe a ser cortada neste caso é a **classe 3**, como é retratado abaixo:

```
>>> classe_cortada
3
```

Listing 7: Valor da variável *classe_cortada*

5.1.2 Retorno sustentável ótimo

Do mesmo modo, verificando pelo console do Python o valor assumido pela variável *Ret*, conclui-se que o retorno ótimo para este exercício é de R$14.711,00 (para $s = 1000$ árvores), que é obtido ao se cortar todas as árvores da terceira classe.

```
>>> Ret
14711.86
```

Listing 8: Valor da variável *Ret*

5.2 Resolução algébrica usando o resultado do Teorema 1

Sendo $k = 2, 3, 4, 5, 6$, faz-se a iteração demandada pela relação (15) como se segue:

$$RT_2 = \frac{p_2 \cdot s}{\frac{1}{g_1}} = \frac{50s}{\frac{1}{0.28}} = 14s$$

$$RT_3 = \frac{p_3 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2}} = \frac{100s}{\frac{1}{0.28} + \frac{1}{0.31}} = 14.711s$$

$$RT_4 = \frac{p_4 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3}} = \frac{150s}{\frac{1}{0.28} + \frac{1}{0.31} + \frac{1}{0.25}} = 13.892s$$

$$RT_5 = \frac{p_5 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + \frac{1}{g_4}} = \frac{200s}{\frac{1}{0.28} + \frac{1}{0.31} + \frac{1}{0.25} + \frac{1}{0.23}} = 13.205s$$

$$RT_6 = \frac{p_6 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + \frac{1}{g_4} + \frac{1}{g_5}} = \frac{250s}{\frac{1}{0.28} + \frac{1}{0.31} + \frac{1}{0.25} + \frac{1}{0.23} + \frac{1}{0.37}} = 14s$$

5.2.1 Classe cortada

O maior valor de de RT_k ocorre quando $k = 3$, então é a classe 3 que deve ser cortada. Concordante com o resultado encontrado na seção anterior.

5.2.2 Retorno sustentável ótimo

Independentemente do número total de árvores s na floresta, o retorno sustentável ótimo neste caso é $14.711s$ (maior valor de RT_k). Este valor é condizente com os R$14.711,86 encontrados com a função *retorno_otimo* no Python, para $s = 1000$.

5.3 Resolução em Python pelo algoritmo Simplex

Para uma análise comparativa com os resultados encontrados anteriormente, se utilizará aqui o método simplex que é um dos métodos mais importantes da área de programação linear (ARENALES et al., 2011), e serve para se determinar a solução ótima em problemas de otimização linear, como é o caso do modelo proposto em (8).

5.3.1 Preparação para a implementação

A implementação do método simplex aqui será baseada na forma padrão retratada a seguir, que é o modelo aceito pelo método $scipy.optimize.linprog(method =' simplex')$, da biblioteca Pyhon chamada *SciPy* (The SciPy community (2021)).

$$
\begin{aligned}
\min_{x} \ & c^T x \\
\text{such that } & A_{ub} x \leq b_{ub}, \\
& A_{eq} x = b_{eq}, \\
& l \leq x \leq u,
\end{aligned}
$$

Figura 2: Formato do problema de programação Linear aceito pela biblioteca

Por padrão, a biblioteca trabalha com valores não negativos para as variáveis de decisão $x_1, x_2, ..., x_n$. Então não será necessário definir as condições de não negatividade na programação em Python, que estão representadas na figura acima por $l \leq x \leq u$, e define limites infirior e superior para as variáveis de decisão.

Para que o modelo em (8) se adeque ao aceito pela biblioteca $SciPy$ devemos reescreve-lo. Este modelo pede que se maximize RT, então no modelo $SciPy$, devemos minimizar o inverso de RT isto é: $min\ (-1) \cdot RT$.

Além disso, nas restrições todas as inequações devem estar na forma $\mathrm{A}_{ub}\mathrm{x} \leq \mathrm{b}_{ub}$ e igualdades na forma $\mathrm{A}_{eq}\mathrm{x} = \mathrm{b}_{eq}$.

Assim, fazendo as alterações necessárias em nosso modelo, considerando os valores de parâmentros de crescimento e preços dados e sendo s=1000, tem-se:

$$
\begin{cases}
Min\ -RT\ =\ -14x_1 - 15.5x_2 - 12.5x_3 - 11.5x_4 - 18.5x_5 - 0x_6 \\
sujeito\ a: \\
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 1000 \\
-0.28x_1 + 0.31x_2 \leq 0 \\
-0.31x_2 + 0.25x_3 \leq 0 \\
-0.25x_3 + 0.23x_4 \leq 0 \\
-0.23x_4 + 0.37x_5 \leq 0 \\
x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6 \geq 0
\end{cases}
\tag{16}
$$

Colocando na forma matricial para que possamos escrever nosso programa tem-se:

Função objetivo:

$$
-RT = \begin{pmatrix} -14 & -15.5 & -12.5 & -11.5 & -18.5 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}
$$

Restrições:
Desigualdade ($A_{ub}x \leq b_{ub}$):

$$\begin{pmatrix} -0.28 & 0.31 & 0 & 0 & 0 & 0 \\ 0 & -0.31 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & -0.25 & 0.23 & 0 & 0 \\ 0 & 0 & 0 & -0.23 & 0.37 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Igualdade ($A_{eq}x = b_{eq}$):

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = (1000)$$

As condições de não negatividade não foram inclusas pois como explicitado elas já são assumidas como padrão pela biblioteca usada.

5.3.2 Implementação em Python

Definindo as variáveis em Python de acordo com as matrizes acima tem-se:

```
s = 1000

negativo_RT = [-14, -15.5, -12.5, -11.5, -18.5, 0]

A_ub = [[-0.28,  0.31,     0,     0,    0, 0],
        [    0, -0.31,  0.25,     0,    0, 0],
        [    0,     0, -0.25,  0.23,    0, 0],
        [    0,     0,     0, -0.23, 0.37, 0]]
b_ub = [0,
        0,
        0,
        0]

A_eq = [[1, 1, 1, 1, 1, 1]]
b_eq = [s]
```

Listing 9: Simplex: definindo as matrizes

Agora, passando as variáveis definidas acima como argumento ao método *linprog* da biblioteca *scipy.optimize*, armazena-se na variável *otimo* os resultados da otimização realizada com o algoritimo simplex.

```python
from scipy.optimize import linprog

otimo = linprog(c=negativo_RT, A_ub=A_ub, b_ub=b_ub, A_eq=A_eq,
                b_eq=b_eq, method='Simplex')
```

Listing 10: Simplex: otimizando

5.3.3 Retorno sustentável ótimo

Agora a variável *otimo* contém todos os resultados da otimização. Pode-se acessar o retorno ótimmo, por meio do comando $otimo.fun$, porém neste caso será retornado um valor negativo, pois trata-se do mínimo de $-RT$, que corresponde ao máximo de $+RT$, então basta que verifiquemos o valor de $-otimo.fun$. Observe:

```python
>>> -otimo.fun.round(2)
14711.86
```

Listing 11: Simplex: retorno ótimo arredondando para 2 casas decimais

Portanto o retorno sustentável ótimo é de R$14.711,86 (quando $s = 1000$), condizente com todos os resultados encontrados anteriormente.

5.3.4 Configuração inicial da floresta que permite um corte sustentável

Pode-se também, por meio da otimização realizada com o método simplex, acessar o conteúdo do vetor x tal que RT é máximo. Isto é, podemos acessar a configuração inicial da floresta que permite um corte sustentável ótimo (para $s = 1000$):

```python
>>> otimo.x.round(0).tolist()
[525.0, 475.0, 0.0, 0.0, 0.0, 0.0]
```

Listing 12: Simplex: vetor x

Como pode-se verificar, neste caso a configuração inicial da floresta é tal que:

$$\mathrm{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 525 \\ 475 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Isto é:

$$x_1 = 525, x_2 = 475, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = 0 \tag{17}$$

6 Valor de p_5 para que seja a classe 5 a ser cortada

6.1 Resolução Algébrica

Como verificado na resolução algébrica feita na seção 5.2, para os preços e parâmetros de crescimento dados, o retorno sustentável ótimo é obtido cortando-se todas as árvores da classe 3, dando um retorno de $RT_3 = 14.711s$.

Portanto, para que seja a classe 5 aquela a ser totalmente removida, o valor de RT_5 deve ser maior que $14.711s$, se $RT_5 = RT_3$ seriamos indiferentes entre cortar totalmente somente as árvores da classe 3 ou somente as da classe 5.

Assim, matematicamente deve-se ter:

$$RT_5 > 14.711s$$
$$\frac{p_5 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + \frac{1}{g_4}} > 14.711s$$
$$\frac{p_5}{\frac{1}{0.28} + \frac{1}{0.31} + \frac{1}{0.25} + \frac{1}{0.23}} > 14.711$$
$$p_5 > 222.81$$

Desta forma, o preço das árvores da classe 5 para que seja esta a ser cortada deve ser maior que R$222,81.

Então p_5 deve ser pelo menos igual a R$222,82.

6.2 Simulação em Python

Neste caso basta fazer testes sucessivos, nos quais se aumenta progrssivamente o valor de p_5 até que RT_5 seja o maior valor dentre todos os RT_k. Ao se incrementar p_5 desta forma, **o primeiro valor de p_5 que faz RT_5 ser máximo é o preço necessário para que a quinta classe seja a que vamos retirar completamente para termos o retorno sustentável ótimo.**

A implementação em Python do processo iterativo descrito é representada a seguir e utiliza a função *retorno_otimo* definida em *Listing 3*, que realiza o processo descrito em (15). Aproveita-se também a lista *precos* definida em *Listing 5*, de modo que *precos*[3] representa p_5.

```python
s = 1000
classe_cortada = -1
while classe_cortada != 5: #testa-se ate que classe_cortada == 5
  precos[3] = precos[3] + 0.001
  __, Ret, classe_cortada = retorno_otimo(classes, s, precos, g)
```

Listing 13: Encontrando p_5 tal que RT_5 é máximo

No algoritmo acima, incrementa-se o valor de p_5 em 0.001 a cada passo até que *classe_cortada* seja igual a 5. Assim, o último valor de $precos[3]$ registrado na memória é o valor de p_5 que faz RT_5 ser máximo.

```
>>> round(precos[3], 2)
222.82
```

Listing 14: Valor de p_5 tal que *classe_cortada* é 5

Portanto o valor de p_5 para que seja a quinta classe a ser cortada é: R$222,82. Valor este que é condizente com a solução algébrica demonstrada anteriormente.

Na *Listing 13* a variável s está definida com o valor 1000, porém alterando-se o valor de s para qualquer outro valor como $s = 10, 100, 1000000$, o valor encontrado de $precos[3]$ é sempre o mesmo e igual a R$222,82.

7 relação entre p_2, p_3, p_4, p_5 e p_6 para que qualquer que seja a classe cortada, teremos retorno sustentável ótimo

Para que tenhamos um retorno sustentável ótimo qualquer que seja a classe a ser cortada, sendo $k = 2, 3, 4, 5, 6$, o retorno oferecido ao se cortar qualquer uma das classes deve ser o mesmo, então:

$$RT_2 = RT_3 = RT_4 = RT_5 = RT_6 \tag{18}$$

Como queremos encontrar a razão entre os preços, podemos estabelecer arbitrariamente o valor de um deles. Então por conveniência, seja $p_2 = 1$ o preço das árvores da classe 2 que dê retorno ótimo.

Cortando a classe k, o retorno ótimo é dado por (14). Assim, para $k = 2$ e $p_2 = 1$, tem-se:

$$RT_2 = \frac{p_2 \cdot s}{\frac{1}{g_1}} = \frac{1 \cdot s}{\frac{1}{0.28}} \therefore RT_2 = 0.28s \tag{19}$$

Substituindo (19) em (18):

$$0.28s = RT_3 = RT_4 = RT_5 = RT_6$$

Dados os valores de g_1, g_2, g_3, g_4 e g_5, temos então:

$$RT_3 = \frac{p_3 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2}} \therefore 0.28s = \frac{p_3 \cdot s}{\frac{1}{0.28} + \frac{1}{0.31}} \therefore p_3 = 1.9$$

$$RT_4 = \frac{p_4 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3}} \therefore 0.28s = \frac{p_4 \cdot s}{\frac{1}{0.28} + \frac{1}{0.31} + \frac{1}{0.25}} \therefore p_4 = 3.02$$

$$RT_5 = \frac{p_5 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + \frac{1}{g_4}} \therefore 0.28s = \frac{p_5 \cdot s}{\frac{1}{0.28} + \frac{1}{0.31} + \frac{1}{0.25} + \frac{1}{0.23}} \therefore p_5 = 4.24$$

$$RT_6 = \frac{p_6 \cdot s}{\frac{1}{g_1} + \frac{1}{g_2} + \frac{1}{g_3} + \frac{1}{g_4} + \frac{1}{g_5}} \therefore 0.28s = \frac{p_6 \cdot s}{\frac{1}{0.28} + \frac{1}{0.31} + \frac{1}{0.25} + \frac{1}{0.23} + \frac{1}{0.37}} \therefore p_6 = 5$$

Isto é:

$$p_2 = 1, p_3 = 1.9, p_4 = 3.02, p_5 = 4.24, p_6 = 5$$

Então a razão entre os preços $p_2 : p_3 : p_4 : p_5 : p_6$ para que qualquer que seja a classe cortada dê retorno sustentável ótimo é:

$$1 : 1.9 : 3.02 : 4.24 : 5$$

8 Quantidade de ávores removidas em cada colheita

8.1 Resolução algébrica

A quantidade r de árvores removidas é dada por:

$$r = y_2 + y_3 + ... + y_n$$

Pela primeira equação do sistema (3), tem-se que:

$$y_2 + y_3 + \cdots + y_n = g_1 x_1$$

Então:

$$r = g_1 \cdot x_1 \tag{20}$$

Usando a expressão (12) de x_1 em função dos parâmetros s e g, e substituindo na relação (20) acima, tem-se:

$$r = g_1 \left[\frac{s}{g_1 \left(\frac{1}{g_1} + \frac{1}{g_2} + ... + \frac{1}{g_{k-1}} \right)} \right]$$

Por fim, a quantidade de árvores removidas pode ser dada por:

$$r = \frac{s}{\frac{1}{g_1} + \frac{1}{g_2} + ... + \frac{1}{g_{k-1}}}$$

Sendo k a classe que será completamente removida.

Para o exemplo prático iniciado na seção 5, a classe removida é a terceira, portanto $k = 3$ na fórmula acima. Então para este problema, temos:

$$r = \frac{s}{\frac{1}{g_1} + \frac{1}{g_2}} = \frac{s}{\frac{1}{0.28} + \frac{1}{0.31}} \therefore r = 0.147s$$

Portanto para este exemplo, como $r = 0.147s$, a quantidade de árvores removidas é sempre 14.7% da quantidade total de árvores na floresta.

Por exemplo, para o caso em que $s = 1000$, a quantidade de árvores removidas em cada colheita é:

$$r = 0.147 \cdot s = 0.147 \cdot 1000$$

$$r = 147$$

8.2 Comparação com a resposta do algoritmo Simplex

Apenas para efeito de comparação, usaremos aqui o valor de x_1 encontrado na seção 5 a partir da implementação do algoritmo simplex como parte da solução ótima (quando $s = 1000$).

Por (17) sabe-se que quando $s = 1000$, $x_1 = 525$. Substituindo em (20), tem-se:

$$r = g_1 \cdot x_1 \therefore r = 0.28 \cdot 525$$

$$r = 147$$

Condizente com o resultado encontrado anteriormente.

9 Verificando numericamente o resultado de otimalidade

9.1 Apresentação

Como visto acima, para que o vetor x determine uma configuração da floresta que permita um corte sustentável, os valores de x_i devem satisfazer as seguintes condições, que são condições necessárias e suficientes para que o objetivo de sustentabilidade seja atingido (além é claro do fato de que $x_i \geq 0$):

$$g_1x_1 \geq g_2x_2 \geq ... \geq g_{n-1}x_{n-1} \geq 0$$

Para o caso do problema em questão, deve-se ter:

$$0.28x_1 \geq 0.31x_2 \geq 0.25x_3 \geq 0.23x_4 \geq 0.37x_5 \geq 0 \tag{21}$$

Assim, não é difícil encontrar configurações da floresta que permitam um corte sustentável. Por exemplo, se $s = 1000$, a seguinte configuração permite um corte sustentável:

$$x_1 = 900, x_2 = 100, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = 0 \tag{22}$$

Já que:

$$0.28 \cdot 900 \geq 0.31 \cdot 100 \geq 0.25 \cdot 0 \geq 0.23 \cdot 0 \geq 0.37 \cdot 0 \geq 0$$

$$252 \geq 31 \geq 0 \geq 0 \geq 0 \geq 0$$

Levando-se em conta a função de retorno financeiro que já assume as condições de corte sustentável, istó é, a função de **retorno sustentável** vista em (5) e aplicando-a ao presente problema, tem-se que a função de retorno sustentável neste caso é dada por:

$$RT = 14x_1 + 15.5x_2 + 12.5x_3 + 11.5x_4 + 18.5x_5 \tag{23}$$

Substituindo os valores em (22) tem-se:

$$RT = 14 \cdot 900 + 15.5 \cdot 100 + 12.5 \cdot 0 + 11.5 \cdot 0 + 18.5 \cdot 0$$

$$RT = \$14.150,00$$

Portanto fazendo um corte sustentável em uma floresta que tem a configuração vista em (22) dá um retorno financeiro de R$ 14.150,00, que não é o maior valor financeiro que pode ser obtido para uma floresta com 1000 árvores. Como visto na discussão sobre o retorno sustentável ótimo, o máximo retorno financeiro que pode ser obtido a partir de um corte sustentável em uma foresta com 1000 árvores é de R$14.711,86.

9.2 Verificando o retorno sustentável para diferentes configurações de florestas

Fazendo o procedimento anterior para algumas configurações de florestas que passam no teste (21), monta-se a seguinte tabela para uma floresta que comporta 1000 árvores:

$Configuracao$	x_1	x_2	x_3	x_4	x_5	x_6	RT
1	350	300	200	100	50	0	$R\$14.125,00$
2	400	300	200	100	0	0	$R\$13.900,00$
3	400	350	200	50	0	0	$R\$14.100,00$
4	450	350	200	0	0	0	$R\$14.225,00$
5	450	400	150	0	0	0	$R\$14.375,00$
6	500	450	50	0	0	0	$R\$14.600,00$
7	510	460	30	0	0	0	$R\$14.645,00$
8 (ótimo)	**526**	**474**	**0**	**0**	**0**	**0**	**R$ 14.711,00**
9	600	400	0	0	0	0	$R\$14.600,00$
10	700	300	0	0	0	0	$R\$14.450,00$
11	800	200	0	0	0	0	$R\$14.300,00$
12	900	100	0	0	0	0	$R\$14.150,00$
13	1000	0	0	0	0	0	$R\$14.000,00$

Dentre os valores testados acima o maior retorno sustentável é conseguido com uma configuração inicial da floresta em que: $x_1 = 526, x_2 = 474, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = 0$. Próxima da configuração ótima encontrada pelo algoritmo simplex na seção 5. As diferenças aqui se devem apenas a questões de aproximação de valores.

9.3 Caso geral

Poderiamos fazer uma tabela que abrangesse o caso geral e que não dependesse do valor de s, nesse caso os números de árvores em cada classe estariam representadas como uma porcentagem da quantidade total de árvores na floresta, sempre verificando se os valores de x_i satisfazem as condições em (21).

Por exemplo, para o caso em que:

$$x_1 = 0.35s, x_2 = 0.3s, x_3 = 0.2s, x_4 = 0.1s, x_5 = 0.05s, x_6 = 0$$

Substituindo na função de retorno sustentável (23), teriamos:

$$RT = 14x_1 + 15.5x_2 + 12.5x_3 + 11.5x_4 + 18.5x_5$$

$$RT = 14 \cdot (0.35s) + 15.5 \cdot (0.3s) + 12.5 \cdot (0.2s) + 11.5 \cdot (0.1) + 18.5 \cdot (0.05s)$$

$$RT = 14.125s$$

Fazendo o mesmo procedimento para diferentes configurações de florestas como porcetagens do total de árvores tem-se obtem-se uma tabela semelhante à anterior:

$Configuracao$	x_1	x_2	x_3	x_4	x_5	x_6	RT
1	$0.35s$	$0.3s$	$0.2s$	$0.1s$	$0.05s$	0	$14.125s$
2	$0.4s$	$0.3s$	$0.2s$	$0.1s$	$0s$	$0s$	$13.9s$
3	$0.4s$	$0.35s$	$0.2s$	$0.05s$	$0s$	$0s$	$14.1s$
4	$0.45s$	$0.35s$	$0.2s$	$0s$	$0s$	$0s$	$14.225s$
5	$0.45s$	$0.4s$	$0.15s$	$0s$	$0s$	$0s$	$14.375s$
6	$0.5s$	$0.45s$	$0.05s$	$0s$	$0s$	$0s$	$14.6s$
7	$0.51s$	$0.46s$	$0.03s$	$0s$	$0s$	$0s$	$14.645s$
8 (ótimo)	**0.526s**	**0.474s**	**0s**	**0s**	**0s**	**0s**	**14.711s**
9	$0.6s$	$0.4s$	$0s$	$0s$	$0s$	$0s$	$14.6s$
10	$0.7s$	$0.3s$	$0s$	$0s$	$0s$	$0s$	$14.45s$
11	$0.8s$	$0.2s$	$0s$	$0s$	$0s$	$0s$	$14.3s$
12	$0.9s$	$0.1s$	$0s$	$0s$	$0s$	$0s$	$14.15s$
13	$1s$	$0s$	$0s$	$0s$	$0s$	$0s$	$14s$

O maior retorno sustentável encontrado nos testes acima (na configuração 8) é condizente com o valor do retorno sustentável ótimo RT_3 encontrado a partir da solução algébrica feita na seção 5, em que que o maior retorno é obtido cortando-se todas as árvores da classe 3.

9.4 Configurções do vetor de corte y

A partir dos valores de x_i que determinam a configuração da floresta que permite um corte sustentável, é possível encontrar qual será a "configuração deste corte", isto é, determinar os valores de y_i.

Com base em (2) e (3) tem-se que:

$$\begin{aligned} y_1 &= 0 \\ y_2 &= g_1x_1 - g_2x_2 \\ y_3 &= g_2x_2 - g_3x_3 \\ &\vdots \\ y_{n-1} &= g_{n-2}x_{n-2} - g_{n-1}x_{n-1} \\ y_n &= g_{n-1}x_{n-1} \end{aligned}$$

Portanto, para o problema em questão:

$$
\begin{aligned}
y_1 &= 0 \\
y_2 &= 0.28x_1 - 0.31x_2 \\
y_3 &= 0.31x_2 - 0.25x_3 \\
y_4 &= 0.25x_3 - 0.23x_4 \\
y_5 &= 0.23x_4 - 0.37x_5 \\
y_6 &= 0.37x_5
\end{aligned}
$$

Com base nisso descubriremos agora a configuração da colheita, isto é, a quantidade de árvores de cada classe que será cortada (y_i) em cada uma das configurações de floresta vistas na seção anterior ($s = 1000$).

$Configuracao$	$y1$	$y2$	$y3$	$y4$	$y5$	$y6$
1	0	5	43	27	5	19
2	0	19	43	27	23	0
3	0	4	59	39	12	0
4	0	18	59	50	0	0
5	0	2	87	38	0	0
6	0	1	127	13	0	0
7	0	0	135	8	0	0
8 (ótimo)	**0**	**0**	**147**	**0**	**0**	**0**
9	0	44	124	0	0	0
10	0	103	93	0	0	0
11	0	162	62	0	0	0
12	0	221	31	0	0	0
13	0	280	0	0	0	0

Os valores da tabela acima foram arredondados.

Como pode-se ver, o retorno sustentável ótimo (*Configuração 8*) é obtido cortando-se todas as árvores da classe 3 e nenhuma árvore de qualquer outra classe. De fato após o período de crescimento, fazendo uma aproximação, haverá 147 árvores na classe 3 (isto é: $x_3 + g_2x_2 = 0 + 0.31 \cdot 475 \backsim 147$). No retorno ótimo, estas 147 árvores da classe 3 são completamente removidas, pois neste caso $y_3 = 147$.

Na configuração *configuração 13* são cortadas todas as 280 árvores da classe 2 (número de árvores na classe 2 após o período de crescimento: $x_2 + g_1x_1 = 0 + 0.28 \cdot 1000 = 280$) e nenhuma árvores de qualquer outra classe. Porém este fato não entra em conflito com o Teorema 1 apresentado, pois a lógica do teorema é a seguinte:

Verdadeiro: "No rendimento ótimo corta-se todas as árvores de somente uma classe e nenhuma árvore das outras classes"

O que não implica que o contrário seja verdadeiro:

Falso: "Ao cortar-se todas as árvores de somente uma classe e nenhuma árvore das outras classes, tem-se rendimento ótimo"

O que pudemos constatar pela tabela é que, pelo menos para o caso deste problema, quando se tem retorno ótimo (*Configuração 8*), corta-se todas as árvores de somente uma classe (neste caso, a classe 3) e nenhuma árvore de qualquer outra classe, estando de acordo com o Teorema 1.

Fazendo para o caso geral da seção 9.3 a constatação é a mesma.

10 Análise crítica e comparação com outros modelos

10.1 Crítica ao modelo

Como todo modelo utilizado teve suas limitações quando comparamos a situação real de um plantio, especialmente deste modelo repara-se que é utilizado preços fixos e não é considerado dinamismo do mercado ou até índices inflacionários como IPCA ou IGPM, pois durante o tempo poderia se ter atualização dos valores para que plantio tivesse uma receita que não perdesse ao menos poder compra devido a inflação.

Outro ponto considerado foram os fatores de época do plantio, clima da região e outros fatores que são determinantes para crescimento do plantio. Determinar taxas fixas de crescimento das árvores também é um aspecto que não considera dinamismo que a plantação pode sofrer devido aos fatores ambientais ao redor, o modelo faz como se plantio fosse um sistema isolado o que pode fazê-lo ser menos preciso.

Considerar nenhuma taxa de mortalidade das árvores acaba sendo algo que também distância mais ainda o modelo com a realidade, considerando fatores externos como praga, uma possível má formação da planta.

10.2 Explorando artigos de outra área

10.2.1 O modelo de Leslie

Um problema que se se assemelha muito com a construção do modelo analisado neste relatório, é a quetão da determinação da estrutura etária estável dos membros de uma população animal. Entre outros autores, a questão da dinâmica populacional em termos de faixas etárias foi bem estudada por P. H. Leslie em (LESLIE, 1945), trabalho no qual ele analisa como fatores tais quais a taxa de fecundidade das fêmeas de uma população e taxa de mortalidade afetam a estrutura etária da população observada em um instante de tempo subsequente.

O modelo discutido por Leslie, serviu de inspiração para o desenvolvimento do modelo de crescimento e corte de árvores que foi discutido neste relatório, como pode ser constatado em USHER (1966). Fazendo um paralelo, a estrutura etária da população animal em tempo futuro seria o equivalente à configuração da floresta após o período de crescimento.

Uma das simplicações que Leslie faz em seu modelo é a de que a população, em todas as faixas etárias, é constituida apenas por fêmeas.

A estrutura etária da população animal no instante de tempo t é definida como o vetor coluna apresentado a seguir, que se assemlha com o vetor x discutido ao longo de todo o texto.

$$\mathrm{n_t} = \begin{pmatrix} n_{0,\ t} \\ n_{1,\ t} \\ \vdots \\ n_{k,\ t} \end{pmatrix} \tag{24}$$

Leslie define também p_i como a probabilidade de uma fêmea pertencente a faixa etária i no instante de tempo t, estar viva na faixa etária $i+1$ no instante de tempo $t+1$. Então, fazendo uma comparação como o modelo de retorno sustentável ótimo da floresta, os valores de p_i do modelo de Leslie se assemelham com os parâmetros de crescimento das árvores g_i.

Porém Leslie não define a probabilidade de uma fêmea pertencente a faixa i permanecer nesta faixa após o periodo de tempo considerado. Isto é, a taxa de mortalidade de cada faixa etária está implicito em nos valores de p_i. Em termos simples, após o periodo de tempo considerado, ou o animal avança para a faixa etária imediatamente superior ou morre.

Ainda é definido a taxa de fecundidade f_i das fêmeas de cada faixa etária i.

Leslie define então, a matriz M, que reúne os efeitos combinados de mortalidade e fertilidade (natalidade) da população:

$$\mathrm{M} = \begin{pmatrix} f_0 & f_1 & \dots & f_{k-1} & f_k \\ p_0 & 0 & \dots & 0 & 0 \\ 0 & p_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & p_{k-1} & 0 \end{pmatrix} \tag{25}$$

Portanto, a distribuição etária da população de fêmeas em $t+1$ é dada por:

$$\begin{pmatrix} f_0 & f_1 & \dots & f_{k-1} & f_k \\ p_0 & 0 & \dots & 0 & 0 \\ 0 & p_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & p_{k-1} & 0 \end{pmatrix} \begin{pmatrix} n_{0,\ t} \\ n_{1,\ t} \\ \vdots \\ n_{k,\ t} \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{k} f_i \cdot n_{i,\ t} \\ p_0 \cdot n_0 \\ p_1 \cdot n_1 \\ \vdots \\ p_{k-1} \cdot n_{k-1,\ t} \end{pmatrix} \tag{26}$$

O termo $\sum_{i=0}^{k} f_i \cdot n_{i,\ t}$ na equação (26) representa a reposição da população, os nascidos que são adicionados à primeira faixa etária. Pode-se fazer um paralelo com deste termo com a expressão $\sum_{i=2}^{n} y_i$, que é a quantidade de mudas que são plantadas a cada colheita, no modelo visto anteriormente.

Concluindo, o modelo apresentado acima é só uma visão breve dos estudos de Leslie nessa área. Ele estava particularmente interessado em estudar a forma com uma população atinge uma distribuição etária estável ao longo do tempo, partindo de uma distribuição etária arbitrária.

10.2.2 Melhorias propostas por Williamson

M. H. Williamson fez algumas proposições (WILLIAMSON, 1959) para aumentar a abrangêcia do modelo proposto por Leslie, como por exemplo considerou que a população animal é formada por machos e fêmeas, propôs um modo de lidar com as diversas formas de sistemas de acasalamento e polimorfismos genéticos.

Para implementar estas mudanças, em seu artigo ele propõe que os escalares n_i, p_i e f_i vistos anteriormente sejam tratados todos como matrizes N_i, P_i e F_i.

Por exemplo n_i torna-se um vetor coluna do seguinte formato $\mathrm{N_i} = \begin{pmatrix} n_{0♂,\ t} \\ n_{0♀,\ t} \end{pmatrix}$
.

Já p_i passa a ser descrito pela seguinte matriz: $\mathrm{P_i} = \begin{pmatrix} p_{i♂} & 0 \\ 0 & p_{i♀} \end{pmatrix}$ para considerar ambos os sexos.

Já a matriz F_i pode adquirir formatos diversos e complexos, a depender do sistema de acasalamento utilizado e questões genéticas.

11 Discussão sobre os parâmetros do modelo

No modelo há variáveis de entradas que são feitas por estimação: p (preço) e g (taxa de crescimento das árvores), neste ponto será pensado métodos ou procedimento para estimar essas variáveis.

O preço há algumas formas que se pode pensar de como estimar desde o que é praticada pela empresa, mas pensando efeitos ao longo do tempo como índice inflacionário geral ou até opção de indexador envolvendo o próprio é possível que realizar conjunto a variável preço uma atualização do através desses índices. Para isso pode ser pela abordagem de indexador geral oferecidos por entidades como a FGV - Fundação Getúlio Vargas (IGPM) ou IBGE – Instituto Brasileiro de Geografia e Estatística (IPCA) , ou o modelo poderia adotar em paralelo a própria mensuração do índice de preço, realizando assim um cálculo médio os preços de maneira histórica e observando como é comportamento do setor que se trabalha e neste tipo de trabalho quanto mais dados, ou seja, conjunto de produtos semelhantes ao do modelo melhor será o índice para que chegue no indexador mais apropriado a ele.

Enquanto a taxa de crescimento da planta, esse tipo de estimativa poderia ser realizado através de aproximações do próprio plantador, pela a tua experiência, mas esse tipo de estimativa seria um tanto grosseria, portanto, um método que pode ser adotado é da própria medição das árvores com algum tempo de observação do teu crescimento, quanto mais ciclos se observar mais estimativa de taxa de crescimento ser aproximará com o que de fato a planta pode crescer em determinado espaço de tempo.

12 Código Fonte e Fonte do PDF

Link do código fonte do colab utilizado na experimentação:
https://colab.research.google.com/drive/1pJLjhM1bEx3vx5CN2OPg12_tvnQ2in_0

Referências

ANTON, H. and RORRES, C. (2012). *Álgebra Linear com Aplicações*. Bookman, Porto Alegre RS.

ARENALES, M., ARMENTANO, V., MORABITO, R., and YANASSE, H. (2011). *Pesquisa Operacional*. Elsevier, Rio de Janeiro RJ.

LESLIE, P. H. (1945). *On the Use of Matrices in Certain Population Mathematics*. Biometrika, Vol. 33, No. 3, pp. 183-212.

The SciPy community (2021). linprog(method='simplex') em https://docs.scipy.org/doc/scipy/reference/optimize.linprog-simplex.html, acesso: 05 de maio de 2021.

USHER, M. B. (1966). *A Matrix Approach to the Management of Renewable Resources, with Special Reference to Selection Forests*. British Ecological Society. Journal of Applied Ecology , Nov., 1966, Vol. 3, No. 2 (Nov., 1966), pp. 355-367.

WILLIAMSON, M. H. (1959). *Some extensions of the use of matrices in population theory*. Bulletin of Mathematical Biophysics 21, 13–17 (1959).