



UnB

**CIC0203 - Computação Experimental -
TA - 2022.2 - Tarefa T6 - Aprimoramento
de uma Simulação**

URL Read-only Overleaf: <https://www.overleaf.com/read/vtvcnysnqkwd>

Marco Antônio Souza de Athayde (masathayde)

Brasília, 2023-01-05 20:35:12Z

Sumário

I	Simulação Computacional	3
1	T6 - Aprimoramento de uma Simulação: Laboratório e Experimento sobre Canibalismo em Simulação Predador-Presa, por Marco Antônio Athayde - masathayde	5
1.1	Introdução	5
1.2	O Fenômeno do Mundo Real	6
1.3	O Laboratório Predador-Presa	7
1.3.1	O Conceito da Simulação	7
1.3.2	O Simulador	8
1.3.2.1	Variáveis Independentes ou de Controle	8
1.3.2.2	Variáveis Dependentes	9
1.3.3	A Hipótese Causal	9
1.3.4	O Código do Simulador	9
1.3.4.1	Modelo	9
1.3.4.2	Grama	10
1.3.4.3	Ovelhas	11
1.3.4.4	Lobos	11
1.4	Os Experimentos Realizados	12
1.5	Discussão e <i>insights</i> preliminares sobre as hipóteses	14
1.6	Conclusão	14
	Bibliografia	17

SUMÁRIO

Lista de Figuras

1.1	Número de peles de lebres e de lince vendidas à Hudson's Bay Company ao longo de 90 anos, ilustrando a variação estável entre as duas espécies, as quais têm relação de predador e presa.	6
1.2	Gráfico de saída da segunda simulação conduzida. O número de lobos chega a 0, o número de ovelhas oscila entre 40 e 60.	13
1.3	Novamente, ao final de 300 passos, o número de lobos é 0. Houve 32 atos de canibalismo durante esta simulação.	13

LISTA DE FIGURAS

Lista de Tabelas

Resumo

Este documento contém o produto da tarefa especificada no título deste documento, conforme as orientações em <https://www.overleaf.com/read/cytswcjsxxqh>.

Parte I

Simulação Computacional

Capítulo 1

T6 - Aprimoramento de uma Simulação: Laboratório e Experimento sobre Canibalismo em Simulação Predador-Presa, por Marco Antônio Athayde - masathayde

1.1 Introdução

Neste trabalho, será usada uma simulação computacional para representar o fenômeno natural da relação entre predadores e presas. A simulação é uma recriação simplificada do fenômeno, na qual parâmetros modificáveis permitem controle do modelo. Será utilizado o framework Mesa, uma ferramenta para desenvolvimento e monitoramento de simulações multiagentes, construído na linguagem Python. O exemplo de simulação *wolf_sheep*, disponibilizado pelos desenvolvedores do Mesa, servirá como fundação do trabalho. Ao modelo adicionará-se uma nova forma de interação entre os agentes: canibalismo entre os predadores. Analisaram-se dados coletados da simulação modificada, para responder a hipótese proposta mais adiante.

O trabalho consistirá das seguintes seções:

1. Descrição do fenômeno real;
2. Apresentação do laboratório de simulações;
3. Apresentação de análises exploratórias dos dados de experimentos realizados com o uso do laboratório;
4. Discussão sobre *insights* obtidos após os experimentos;
5. Conclusões.

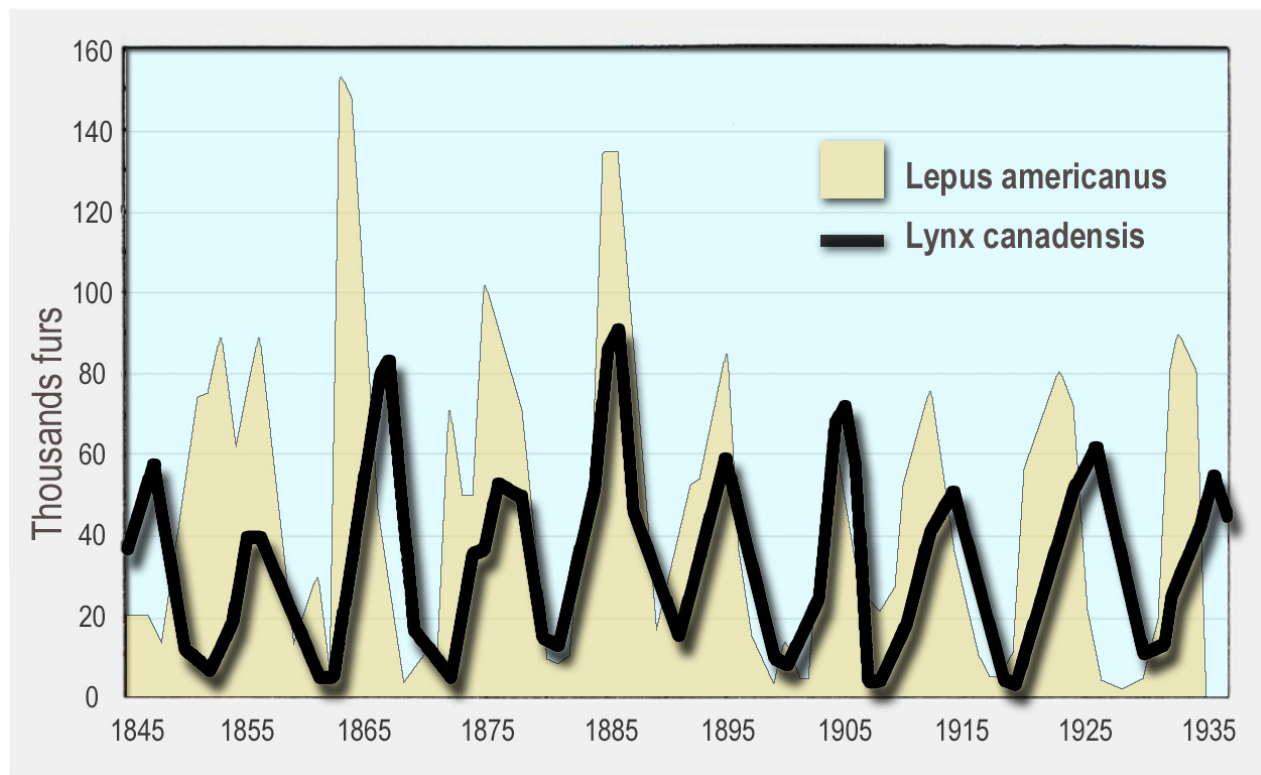


Figura 1.1: Número de peles de lebres e de linces vendidas à Hudson's Bay Company ao longo de 90 anos, ilustrando a variação estável entre as duas espécies, as quais têm relação de predador e presa.

1.2 O Fenômeno do Mundo Real

Predação ocorre quando uma espécie alimenta-se de outra. A espécie que se alimenta é o predador, enquanto a que serve de alimento é a presa. Essa interação tem efeito na quantidade de indivíduos em cada espécie. Em várias instâncias dessa relação, identifica-se um equilíbrio dinâmico dos números das espécies participantes ao longo do tempo, o qual pode ser observado no gráfico a seguir.

O cientista Alfred Lotka estudou vários fenômenos naturais, em busca de modelá-los por meio de equações matemáticas, entre eles a predação - (LOTKA, 1956). O modelo Lotka-Volterra descreve equações diferenciais que descrevem o número de indivíduos em duas espécies envolvidas em uma relação de predação, em função de ambas. Outro trabalho importante foi desenvolvido por C. S. Holling em (HOLLING, 1959), estendendo o modelo Lotka-Volterra para considerar uma característica chamada resposta funcional, a qual descreve a proporção de consumo de um predador como função da densidade de alimento disponível.

Juntamente com outros, os trabalhos acima servem como base para o estudo moderno sobre modelos de predação e simulações, estando entre os mais citados dentro do conjunto

de artigos coletados durante a pesquisa bibliográfica e, subsequentemente, usados na análise bibliométrica.

1.3 O Laboratório Predador-Presa

A atividade descrita neste trabalho consiste em simulação computacional de uma interação predatória entre duas espécies, seguindo um modelo do fenômeno real, apresentado um conjunto de regras de comportamento tanto determinísticas quanto aleatórias, com o intuito de representar de forma simples, abstraindo-se um conjunto significativo de fatores.

Embora seja bem mais simples do que a realidade a qual busca retratar, a simulação oferece uma oportunidade para coletar dados de forma controlada, com a possibilidade de manipulação de parâmetros, para avaliar variadas situações. Caso o modelo tenha um grau satisfatório de correção, é possível inferir conhecimento do mundo real, mesmo a partir de uma simulação simplificada. Essa conveniência serve como justificativa para o presente estudo.

1.3.1 O Conceito da Simulação

Este trabalho foi feito com o framework de simulação multiagente Mesa, desenvolvido em Python. O site principal do módulo pode ser acessado na página da Internet em ([PROJECT MESA, 2021](#)).

A simulação utilizada no trabalho, *wolf_sheep*, é parte do pacote de exemplos disponibilizados em um repositório GitHub da organização desenvolvedora.

Em oposição ao mundo real, o tempo em na simulação Mesa usada no trabalho avança em passos discretos. A cada passo, o estado dos agentes é atualizado simultaneamente. Ademais, o comportamento dos agentes é uma simplificação do mundo real, portanto tais agentes se comportam de forma relativamente previsível e agem de acordo com regras pré-determinadas, ditadas pelos desenvolvedores. Assim, inevitavelmente, um número de suposições e omissões sobre o fenômeno foram feitas.

A simulação em si representa uma relação predador-presa entre 3 agentes. Lobos, ovelhas e grama são as entidades presentes, sendo que lobos se alimentam de ovelhas, e ovelhas se alimentam de grama. Na simulação, lobos e ovelhas reproduzem-se assexualmente, o qual pode ocorrer a cada passo da simulação, de acordo com uma probabilidade determinada pelo controlador. Quando isso ocorre, a quantidade de energia do agente reprodutor é dividida entre ele e seu filho. Gramas regeneram-se independentemente ao longo do tempo.

O mundo da simulação é representado em uma grade bidimensional, a qual forma um conjunto de espaços quadrados distintos. Cada agente da simulação ocupa somente um quadro, embora um quadro pode acomodar vários agentes. Os agentes se deslocam na grade um quadro por vez, escolhendo um quadro adjacente aleatoriamente a cada passo.

Lobos e ovelhas possuem uma quantidade de energia, representada por um número natural. A cada passo, cada um perde um ponto de energia. Quando a quantidade de energia

chega a zero, o lobo ou ovelha em questão morre e é removido da simulação. Ovelhas recuperam energia consumindo grama, e lobos recuperam energia consumindo ovelhas que estejam ocupando o mesmo espaço em que se encontra.

As características acima descrevem o modelo padrão oferecido como exemplo pelos desenvolvedores do Mesa. Neste trabalho, a simulação foi alterada para incluir uma nova forma de interação: canibalismo entre os lobos. Caso cheguem a um nível baixo de energia, um lobo poderá consumir outro lobo, desde que o lobo a ser consumido tenha um nível de energia ainda menor.

1.3.2 O Simulador

O framework Mesa apresenta uma interface gráfica, acessada por meio de um navegador, na qual parâmetros definidores da simulação podem ser alterados, com o uso de objetos gráficos como controladores deslizantes e botões.

1.3.2.1 Variáveis Independentes ou de Controle

A seguir, estão listada as variáveis independentes ou de controle, as quais podem ser manipuladas pelo usuário na interface gráfica:

Grass: Grama. Presença ou não de grama na simulação. Caso não, as ovelhas não se alimentam de grama e nunca perdem energia.

Grass Regrowth Rate: Taxa de regeneração da grama. Quantos passos até um espaço com grama amadurecer a ponto de ser consumível por ovelhas.

Initial Sheep Population: População inicial de ovelhas

Sheep Reproduction Rate: Taxa de reprodução de ovelhas: Representa a chance de uma ovelha reproduzir-se a cada passo

Initial Wolf Population: População inicial de lobos

Wolf Reproduction Rate: Taxa de reprodução de lobos: Representa a chance de um lobo reproduzir-se a cada passo

Wolf Gain From Food: Ganho de energia de um lobo quando se alimenta de uma ovelha.

Sheep Gain From Food: Ganho de energia de uma ovelha quando se alimenta de grama.

Cannibalism Threshold: Nível de energia abaixo do qual lobos passam a considerar canibalismo.

Wolf Gain From Cannibalism: Ganho de energia de um lobo quando se alimenta de outro lobo.

1.3.2.2 Variáveis Dependentes

A seguir, estão listada as variáveis dependentes, as quais são coletadas pelo simulador e retratadas na interface gráfica para visualização:

Wolves: Quantidade lobos vivos.

Sheep: Quantidade ovelhas vivas.

Grass: Quantidade de espaços com grama madura.

Cannibalism Occurrences: Número de vezes que houve canibalismo.

1.3.3 A Hipótese Causal

Há uma quantidade de fatores presentes no fenômeno real omitidos na simulação. Com o propósito de manter a simulação simples, foi adicionado somente um parâmetro a mais, como objetivo de explorar uma situação peculiar, no contexto de predação. Com essa intenção, formulou-se uma hipótese sobre a influência desse fator na simulação.

A hipótese do trabalho está relacionada ao fenômeno de canibalismo. Ao introduzir essa interação na simulação, é feita a seguinte declaração: A existência de canibalismo em um sistema predador-presa pode servir como fator de recuperação em situações extremas, em que a quantidade de presa se torna escassa. Dessa forma, o canibalismo poderia impedir a extinção da espécie predadora em tais cenários, restaurando o equilíbrio do sistema.

Dentro do *dataset* reunido na tarefa anterior, estão presentes alguns trabalhos que exploram os efeitos do canibalismo em sistemas de predação, servindo como suporte à hipótese do trabalho. Em (ZHANG; CHEN; LI, 2019), os autores propõem que uma taxa alta de canibalismo pode influenciar o sistema em direção a um equilíbrio positivo estável, embora um aumento dessa taxa pode ter o efeito contrário. Em (BISWAS; CHATTERJEE; CHATTOPADHYAY, 2015), a partir de um estudo de modelo, os autores sugerem que canibalismo pode ajudar no controle de disseminação de doenças em uma população de predadores e ajudar a preservar o equilíbrio do sistema.

1.3.4 O Código do Simulador

1.3.4.1 Modelo

Listagem de Código 1.1: Variáveis iniciais do modelo

```
18 class WolfSheep(mesa.Model):
19     """
20     Wolf-Sheep Predation Model
21     """
22
23     height = 20
24     width = 20
25
26     initial_sheep = 100
27     initial_wolves = 50
```



```

28
29     sheep_reproduce = 0.04
30     wolf_reproduce = 0.05
31
32     wolf_gain_from_food = 20
33
34     grass = False
35     grass_regrowth_time = 30
36     sheep_gain_from_food = 4

```

Listagem de Código 1.2: Variáveis relacionadas a canibalismo introduzidas no modelo

```

86     # Nível de energia para ativar canibalismo.
87     self.cannibalism_threshold = cannibalism_threshold
88     self.wolf_gain_from_cannibalism = wolf_gain_from_cannibalism
89
90     # Lista de lobos canibalizados a ser removida após um passo da simulação
91     self.wolves_eaten_by_wolves = []
92     self.cannibalism_occurrence = 0

```

Listagem de Código 1.3: Instruções feitas a cada passo da simulação pelo modelo

```

144     self.schedule.step()
145
146     # Remoção de lobos canibalizados
147     for wolf in self.wolves_eaten_by_wolves:
148         self.grid.remove_agent(wolf)
149         self.schedule.remove(wolf)
150         self.cannibalism_occurrence += 1
151     self.wolves_eaten_by_wolves = []
152
153     # collect data
154     self.datacollector.collect(self)
155     if self.verbose:
156         print(
157             [
158                 self.schedule.time,
159                 self.schedule.get_type_count(Wolf),
160                 self.schedule.get_type_count(Sheep),
161                 self.schedule.get_type_count(GrassPatch, lambda x: x.fully_grown),
162             ]
163         )

```

Listagem de Código 1.4: Variáveis dependentes monitoradas pelo modelo

```

96     self.datacollector = mesa.DataCollector(
97         {
98             "Wolves": lambda m: m.schedule.get_type_count(Wolf),
99             "Sheep": lambda m: m.schedule.get_type_count(Sheep),
100             "Grass": lambda m: m.schedule.get_type_count(
101                 GrassPatch, lambda x: x.fully_grown
102             ),
103             "Cannibalism Occurrences": lambda m: m.cannibalism_occurrence,
104         }
105     )

```

1.3.4.2 Grama

Listagem de Código 1.5: Rotina padrão de uma instância de grama

```

117     def __init__(self, unique_id, pos, model, fully_grown, countdown):
118         """
119         Creates a new patch of grass
120
121         Args:
122             grown: (boolean) Whether the patch of grass is fully grown or not
123             countdown: Time for the patch of grass to be fully grown again
124         """
125         super().__init__(unique_id, model)
126         self.fully_grown = fully_grown
127         self.countdown = countdown
128         self.pos = pos
129
130     def step(self):
131         if not self.fully_grown:
132             if self.countdown <= 0:

```

```
133         # Set as fully grown
134         self.fully_grown = True
135         self.countdown = self.model.grass_regrowth_time
136     else:
137         self.countdown -= 1
```

1.3.4.3 Ovelhas

Listagem de Código 1.6: Rotina de alimentação de uma ovelha

```
25     if self.model.grass:
26         # Reduce energy
27         self.energy -= 1
28
29         # If there is grass available, eat it
30         this_cell = self.model.grid.get_cell_list_contents([self.pos])
31         grass_patch = [obj for obj in this_cell if isinstance(obj, GrassPatch)][0]
32         if grass_patch.fully_grown:
33             self.energy += self.model.sheep_gain_from_food
34             grass_patch.fully_grown = False
35
36         # Death
37         if self.energy < 0:
38             self.model.grid.remove_agent(self)
39             self.model.schedule.remove(self)
40             living = False
```

Listagem de Código 1.7: Rotina de reprodução de uma ovelha

```
42     if living and self.random.random() < self.model.sheep_reproduce:
43         # Create a new sheep:
44         if self.model.grass:
45             self.energy /= 2
46             lamb = Sheep(
47                 self.model.next_id(), self.pos, self.model, self.moore, self.energy
48             )
49             self.model.grid.place_agent(lamb, self.pos)
50             self.model.schedule.add(lamb)
```

1.3.4.4 Lobos

Listagem de Código 1.8: Rotina de alimentação de um lobo

```
73     # If there are sheep present, eat one
74     x, y = self.pos
75     this_cell = self.model.grid.get_cell_list_contents([self.pos])
76     sheep = [obj for obj in this_cell if isinstance(obj, Sheep)]
77     if len(sheep) > 0:
78         sheep_to_eat = self.random.choice(sheep)
79         self.energy += self.model.wolf_gain_from_food
80
81     # Kill the sheep
82     self.model.grid.remove_agent(sheep_to_eat)
83     self.model.schedule.remove(sheep_to_eat)
```

Listagem de Código 1.9: Rotina de canibalismo de um lobo

```
84     # Calso não haja ovelhas e nível de energia esteja baixo, poderá haver canibalismo
85     elif self.energy <= self.model.cannibalism_threshold:
86         wolves = [obj for obj in this_cell if isinstance(obj, Wolf)]
87         for wolf in wolves:
88             # Lobo não pode comer a si mesmo, nem pode comer um que tenha mais energia.
89             if id(self) != id(wolf) and wolf.alive and self.energy >= wolf.energy:
90                 self.energy += self.model.wolf_gain_from_cannibalism
91                 wolf.alive = False
92                 # Programar a remoção do lobo que foi comido
93                 self.model.wolves_eaten_by_wolves.append(wolf)
94                 break
```

A interação de canibalismo é introduzida no código de comportamento do agente "Wolf", na descrição do passo de atualização da classe. O código correspondente só é executado caso o agente não tenha realizado a interação original, isto é, o lobo não tenha consumido ovelhas.

Além disso, o código testa a condição de que o lobo a ser potencialmente consumido tenha nível de energia menor, na linha 89 em . A condição ilustra a ideia de que um lobo com menos energia não seria capaz de subjugar outro com mais energia.

1.4 Os Experimentos Realizados

Para testar a razoabilidade da hipótese, foi usada a interface gráfica do simulador para a condução de alguns experimentos iniciais, com o intuito de observar os efeitos da adição das novas variáveis à simulação, sem um plano rigoroso de análise.

Na definição da hipótese, afirmou-se que a existência de canibalismo poderia restaurar o sistema de uma situação de extinção. Para testá-la, primeiro conduziram-se simulações sem incluir a nova interação. Essas instâncias servirão como dados de controle.

Utilizam-se os parâmetros padrões da simulação, listados a seguir:

1. Grass Regrowth: 20 frames
2. Initial Sheep Population: 100
3. Initial Wolf Population: 50
4. Sheep Reproduction Rate: 0.04
5. Wolf Reproduction Rate: 0.05
6. Sheep Gain From Food: 4
7. Wolf Gain From Food: 20
8. Cannibalism Threshold: -1 (Ou seja, canibalismo está desativado)
9. Cannibalism Energy Gain: 10

A seguir, a simulação é acompanhada por 300 passos. Considerando cinco simulações com esses parâmetros, notou-se que, após 300 passos, a população de lobos é extinta. Somente em uma das simulações, a população de lobos permaneceu viva. Em duas simulações, ambas as espécies foram extintas.

Em seguida, a simulação é alterada com a introdução de canibalismo caso o nível de energia de lobos caia para 3, com um ganho de energia de 10, ou seja, metade do ganho de energia provido por ovelhas. Em cinco simulações, a população de lobos foi extinta. Em 3 simulações, a população de ovelhas também foi extinta.

Nos experimentos com canibalismo, observou-se a presença do ato por meio da variável dependente adicionada ao modelo - "Cannibalism occurrences- indicando que a simulação foi alterada com êxito.

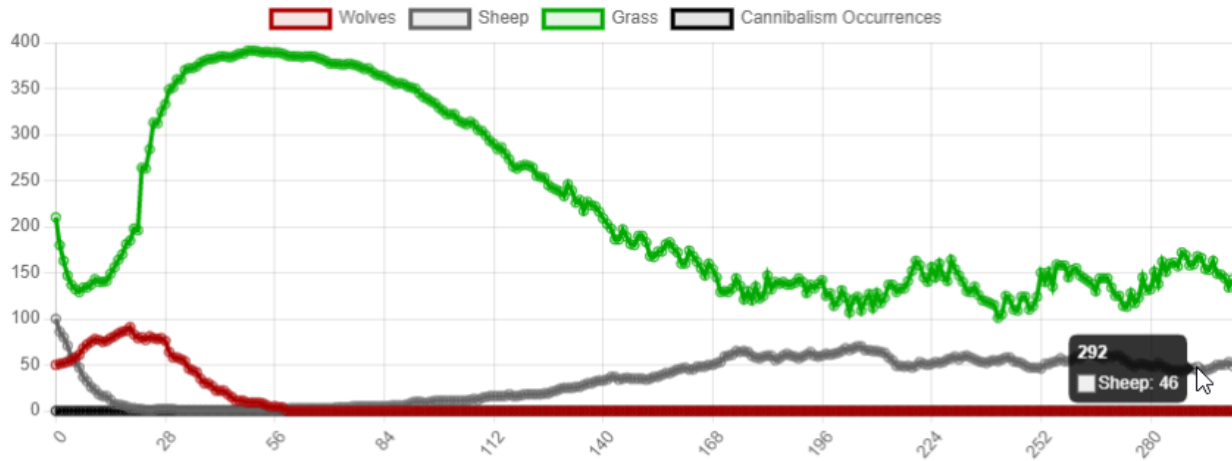


Figura 1.2: Gráfico de saída da segunda simulação conduzida. O número de lobos chega a 0, o número de ovelhas oscila entre 40 e 60.

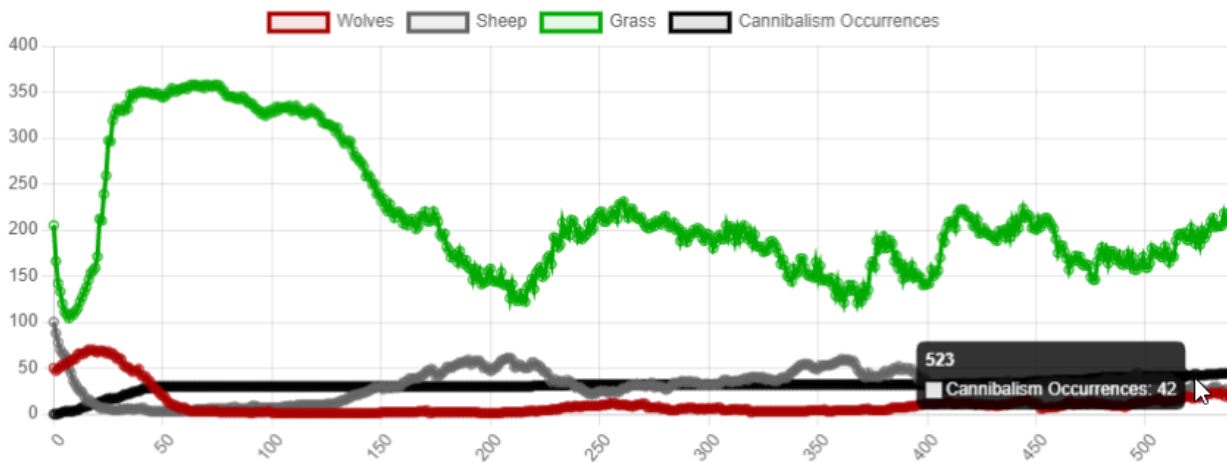


Figura 1.3: Novamente, ao final de 300 passos, o número de lobos é 0. Houve 32 atos de canibalismo durante esta simulação.

1.5 Discussão e *insights* preliminares sobre as hipóteses

Nos experimentos iniciais, com os parâmetros padrões da simulação, após um determinado número de passos, em uma análise inicial, não observou-se uma situação em que a hipótese causal apresentada tenha sido comprovada. Nas condições escolhidas, a introdução do canibalismo não pareceu ter efeito significativo nos experimentos, e, em nenhum momento, ocorreu um quadro de recuperação após um certo número de indivíduos tenha diminuído a um ponto extremo.

No entanto, os resultados iniciais não necessariamente indicam que a hipótese está completamente incorreta. O canibalismo é só um dos fatores de influência, e não é esperado que seja capaz de completamente decidir o estado final da simulação. De fato, nas instâncias registradas nos experimentos iniciais, percebeu-se um número maior de cenários em que a população de ovelhas foi extinta, em relação às simulações sem canibalismo. Esse fato pode ser consequência do canibalismo estendendo a longevidade da população do lobos, a qual poderia caçar ovelhas por mais tempo, resultando na extinção da espécie herbívora.

Considerando essas observações, evidenciou-se a necessidade de criação de novas variáveis de saída, para melhor detectar as influências das características novas colocadas na simulação. Embora não tenha o efeito extremo esperado de reversão de extinção, ainda resta verificar com mais precisão as ramificações da introdução do canibalismo ao sistema.

1.6 Conclusão

Com este trabalho, houve um contato inicial com o framework Mesa e a simulação de sistema predador-presa apresentado como exemplo. A alteração proposta pelo trabalho foi introduzida com sucesso, com a possibilidade de controle por meio da interface gráfica do próprio framework.

Uma hipótese causal acerca da alteração feita foi sugerida. No começo do trabalho, era esperado uma influência significativa e imediatamente visível da introdução de canibalismo entre predadores, a ponto de reverter quadros de extinção, em relação a simulações sem esse fator novo. Essa hipótese baseou-se em trabalhos encontrados na pesquisa bibliográfica feita anteriormente, nos quais o canibalismo aparece como um elemento ecologicamente benéfico, em certas circunstâncias.

Quanto à simulação estudada, há algumas dúvidas em relação à sua capacidade de modelar o fenômeno real satisfatoriamente. Posto que simplificação e abstração são esperadas do modelo, há possibilidade de torná-lo mais preciso e próximo da realidade com algumas alterações pequenas. Em particular, talvez a remoção de aleatoriedade no movimento dos agentes, com a introdução de impacto de eventos meteorológicos e aspectos geográficos, possa ocasionar uma melhora no modelo, sem grande perda de simplicidade.

Em relação ao comportamento canibal colocado, também vislumbram-se possíveis alterações em sua lógica de funcionamento, com o intuito de melhor refletir a realidade do fenômeno. No mundo real, canibalismo ocorre também quando um indivíduo consome o cadáver de outro

da mesma espécie, algo que não está presente na simulação, visto que indivíduos mortos são removidos, perdendo a capacidade de influenciar o sistema.

Neste trabalho, um número relativamente pequeno de simulações foi feita, com o propósito de adquirir intuição sobre os passos de uma pesquisa centrada em uma simulação, desde seu funcionamento e construção, até coleta de dados e análise. Em próximos trabalhos, um número maior de simulações será realizados, com dados mais refinados, para melhor estudar o fenômeno escolhido.

Bibliografia

- BISWAS, Santosh; CHATTERJEE, Samrat; CHATTOPADHYAY, Joydev. Cannibalism may control disease in predator population: result drawn from a model based study. en. *Mathematical Methods in the Applied Sciences*, v. 38, n. 11, p. 2272–2290, jul. 2015. ISSN 01704214. DOI: [10.1002/mma.3220](https://onlinelibrary.wiley.com/doi/10.1002/mma.3220). Disponível em: <<https://onlinelibrary.wiley.com/doi/10.1002/mma.3220>>. Acesso em: 3 jan. 2023. Citado na p. 9.
- HOLLING, C. S. Some Characteristics of Simple Types of Predation and Parasitism. en. *The Canadian Entomologist*, v. 91, n. 7, p. 385–398, jul. 1959. ISSN 1918-3240, 0008-347X. DOI: [10.4039/Ent91385-7](https://www.cambridge.org/core/journals/canadian-entomologist/article/abs/some-characteristics-of-simple-types-of-predation-and-parasitism1/9E1E7D2CCC314766A424680444F4EA9F). Disponível em: <<https://www.cambridge.org/core/journals/canadian-entomologist/article/abs/some-characteristics-of-simple-types-of-predation-and-parasitism1/9E1E7D2CCC314766A424680444F4EA9F>>. Acesso em: 5 dez. 2022. Citado na p. 6.
- LOTKA, Alfred J. *Elements of mathematical biology*. Dover Publications, 1956. Citado na p. 6.
- PROJECT MESA. *Mesa: Agent-based modeling in Python 3+ — Mesa .1 documentation*. Disponível em: <<https://mesa.readthedocs.io/en/latest/>>. Acesso em: 20 jul. 2021. Citado na p. 7.
- ZHANG, Fengqin; CHEN, Yuming; LI, Jianquan. Dynamical analysis of a stage-structured predator-prey model with cannibalism. en. *Mathematical Biosciences*, v. 307, p. 33–41, jan. 2019. ISSN 00255564. DOI: [10.1016/j.mbs.2018.11.004](https://linkinghub.elsevier.com/retrieve/pii/S0025556418305170). Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0025556418305170>>. Acesso em: 3 jan. 2023. Citado na p. 9.