Python para Modelagem Baseada em Agentes `aula 0`

Furtado, Bernardo Alves

May 12, 2020

Menu do dia: boas-vindas, apresentações, curso, instalaçoes, interfaces, HelloWorld!

Introdução
- Apresentação

Desafio
- Algoritmo

`Hands-on`
- Instalações
- Exercícios iniciais

Bernardo Alves Furtado

- Pesquisador produtividade CNPq 2014–
- Ph.D Utrecht University, 2009
- Co-tutorship UFMG Dr. Economia Regional
- Arquiteto, urbanista, mestre em Geografia/GIS
- Professor (1988/2003/2006–)
- Ipea: 2009–2013 Políticas urbanas
- Ipea: 2014– Sistemas complexos e ABM

Contatos e links

- bernardo.furtado@ipea.gov.br
- researchgate.net/profile/Bernardo_Furtado
- GitHub/BAFurtado/PYthon4ABMIpea
- https://sites.google.com/view/bernardo-alves-furtado/home

Alun@s

- Afiliação/formação
- Experiências/interesses
- Atuação recente

Objetivos curso

- Plano de Ensino
- Operacionalização `Python`
- Classes
- Modelos baseados em agentes
- Entregas

Python

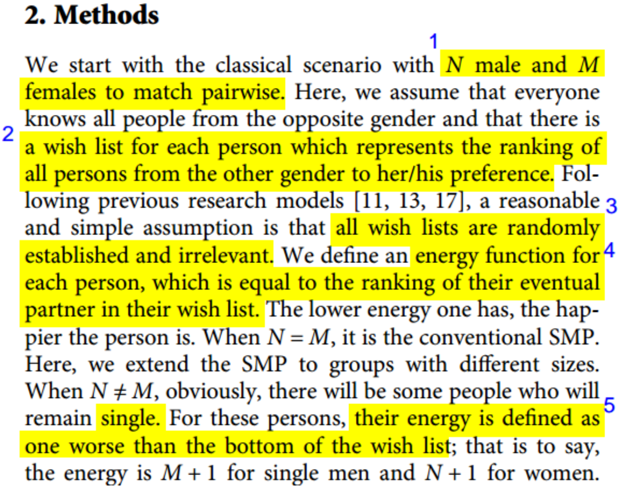
- Estruturas: listas, dicionários, files
- Condicionantes e operadores
- Loops
- Bibliotecas. Operacionalização
- Persistência. Saídas e leituras
- Funções
- Classes. OOP
- Modelagem baseada em agentes
- Exemplos

Instability in the Stable Marriage Problem

Problema original [3]
https://www.hindawi.com/journals/complexity/2018/7409397/

- Método

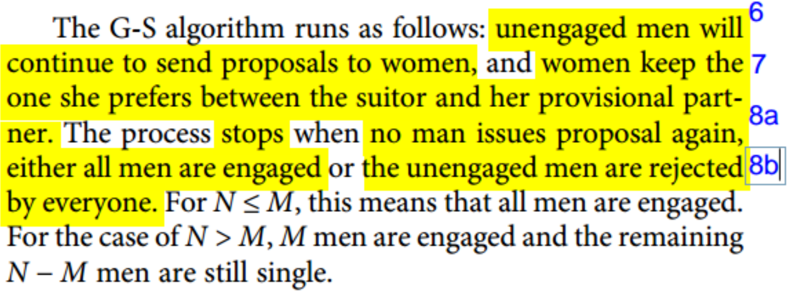
Método I



2. Methods

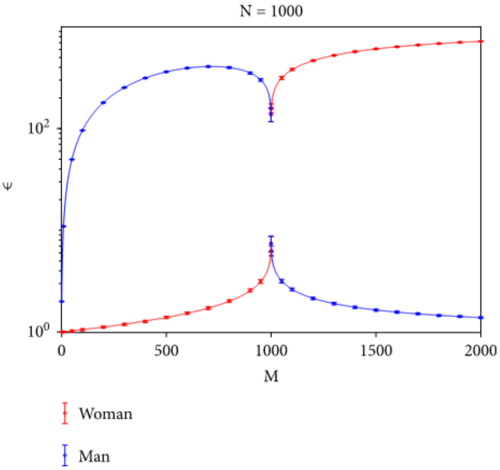
We start with the classical scenario with 1 N male and M females to match pairwise. Here, we assume that everyone knows all people from the opposite gender and that there is 2 a wish list for each person which represents the ranking of all persons from the other gender to her/his preference. Following previous research models [11, 13, 17], a reasonable 3 and simple assumption is that all wish lists are randomly established and irrelevant. We define an energy function for 4 each person, which is equal to the ranking of their eventual partner in their wish list. The lower energy one has, the happier the person is. When $N = M$, it is the conventional SMP. Here, we extend the SMP to groups with different sizes. When $N \neq M$, obviously, there will be some people who will remain single. For these persons, their energy is defined as 5 one worse than the bottom of the wish list; that is to say, the energy is $M + 1$ for single men and $N + 1$ for women.

Método II



The G-S algorithm runs as follows: unengaged men will continue to send proposals to women, and women keep the one she prefers between the suitor and her provisional partner. The process stops when no man issues proposal again, either all men are engaged or the unengaged men are rejected by everyone. For $N \leq M$, this means that all men are engaged. For the case of $N > M$, M men are engaged and the remaining $N - M$ men are still single.

Método III



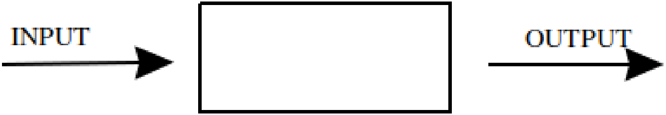
Python: referências básicas

- Think Python [2]
- greenteapress.com/wp/think-python-2e/
- Think Complexity [1]
- greenteapress.com/wp/think-complexity-2e/

PyCharm Community e Anaconda

- **PyCharm**: ide, interface, ambiente, RStudio
- https://www.jetbrains.com/pycharm/download/
- **Conda**: `python` e suas bibliotecas, libraries
- https://www.anaconda.com/download

Program, Script, Software, App



Running python file: hello.py

- Console
  - $ python
  - >>> print('Hello world')
- Terminal
  - $ python hello.py
  - Hello world
- # Do it!

Noções: int, str, float

Digite

- type(5)
- type('5')
- int('5')
- print(5 + 5)
- print('5' + '5')

input, print, variable assignment

Digite

- x = int(input('Entre um número:')
- print('o número é':, x)

Tente: `soma.py`

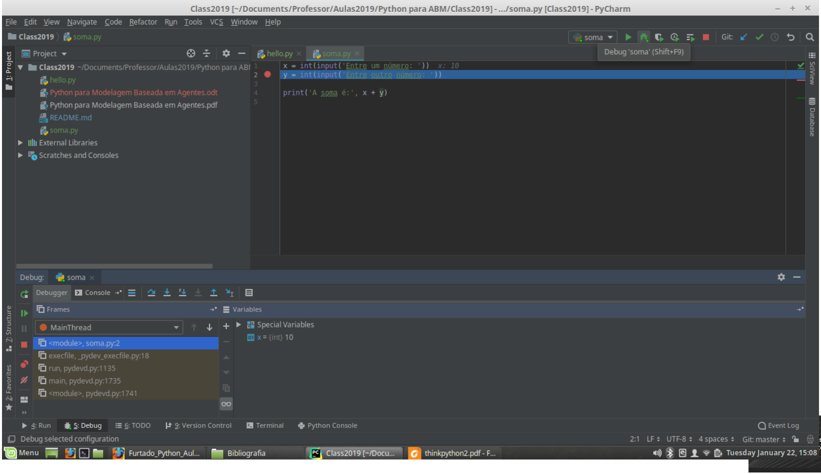
Alguns exercícios

- Descubra esses operadores no console:
- +, -, *, /, **
- Note: order of precedence – PEMDAS
  - Parentheses, Exponentiation, Multiplication, Division, Adition, Subtraction. Da esquerda para a direita
- Quanto é: $(25 * (2 + 23)/54)^2$
- Quanto é: $5 * k$
- minute $= 60$
- Quanto é: $6 * minute$

Floor division and modulus

- Quantas horas são 200 minutos?
- `floor_modulus.py`

Debugging in PyCharm



Class2019 [~/Documents/Professor/Aulas2019/Python para ABM/Class2019] - .../soma.py [Class2019] - PyCharm

```
x = int(input('Entre um número: '))  x: 10
y = int(input('Entre outro número: '))

print('A soma é:', x + y)

```

```
x = {int} 10
```

```
<module>, soma.py:2
execfile, _pydev_execfile.py:18
run, pydevd.py:1135
main, pydevd.py:1735
<module>, pydevd.py:1741
```

Debug selected configuration

Python Challenge 0

- `http://www.pythonchallenge.com/`
- What is the address of the page for Challenge 1?

Exercicios

- Leia o Chapter 1 and 2 do Think Python
- Teste o console, teste o script

Referências I

[1] Allen B. Downey. *Think Complexity: Complexity Science and Computational Modeling*. O'Reilly Media, Sebastopol, CA, 1 edition edition, March 2012.

[2] Allen B. Downey. *Think Python*. O'Reilly Media, United States of America, 2012.

[3] Gui-Yuan Shi, Yi-Xiu Kong, Bo-Lun Chen, Guang-Hui Yuan, and Rui-Jie Wu. Instability in Stable Marriage Problem: Matching Unequally Numbered Men and Women. *Complexity*, 2018:5, 2018.