



## PAC 3

### Presentació

Tercera activitat d'avaluació continuada del curs. En aquesta PAC es pretén conèixer i desenvolupar sistemes multiagent.

### Objectius

L'objectiu d'aquesta PAC és conèixer el funcionament d'un entorn de desenvolupament de sistemes multi-agent. En concret es treballarà amb l'entorn Mesa de Python (<https://github.com/projectmesa/mesa>). Es proporciona la implementació d'un sistema multi-agent i es demana l'anàlisi i la implementació de canvis en el sistema.

### Solució de la PAC

Analitzeu els arxius dins de l'arxiu **Schelling.zip**. Es tracta d'una simulació del model de segregació de Schelling, que descriu com una població tendeix a agrupar-se en funció d'una determinada característica compartida. El problema comença amb dos tipus d'agents: els blaus i els vermells. Aquests agents estan contents si un determinat nombre dels seus 8 possibles veïns són del seu mateix tipus. En cada iteració del model, els agents que no estan contents es mouen aleatòriament a una casella buida. El model continua executant-se fins que tots els agents estan contents. La Figura 1 mostra l'execució de la simulació.

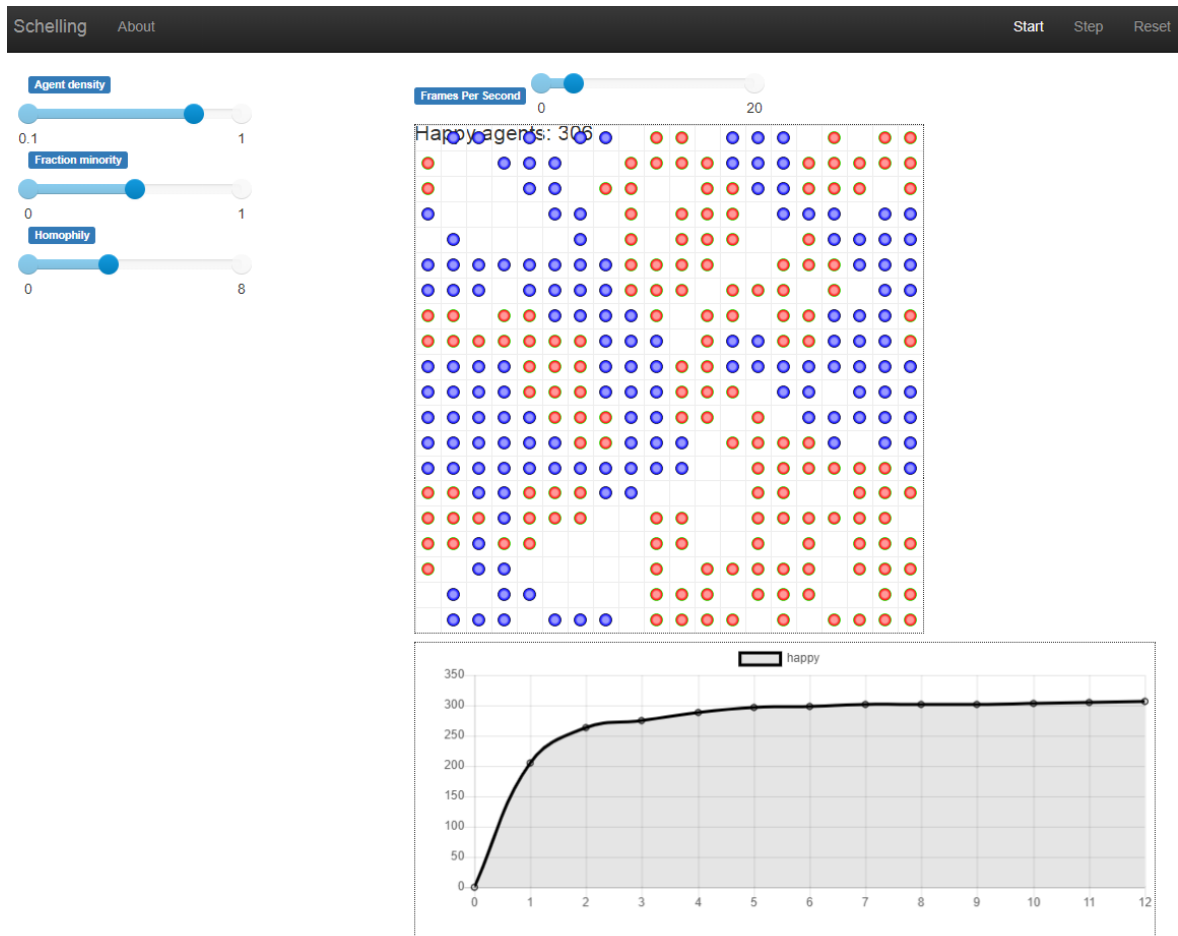


Figura 1. Simulació de l'exemple Schelling del projecte mesa.

En l'arxiu Schelling.zip podeu trobar els següents arxius:

- run.py: conté la comanda d'execució de la simulació en Python (\$python run.py).
- model.py: conté la definició del model, dels paràmetres de simulació, dels agents i de les accions realitzades en cada iteració del model.
- server.py: conté la definició de la interfície gràfica per a la simulació, paràmetres de la simulació, valors a mostrar, gràfiques, etc (veure Figura 1).



## Exercici 1

En aquest exercici se us demana que analitzeu el problema que se us presenta:

1.- Justifiqueu de quin tipus són els agents que apareixen en aquest exemple (reactius o deliberatius).

Tenim 2 tipus agents en aquest sistema. Aquests agents són de tipus reactiu, ja que no tenen la capacitat de comunicar-se entre ells i només reaccionen a les dades que reben dels seus sensors.

2.- Identifiqueu i expliqueu les variables d'aquests agents i les accions que poden desenvolupar.

Variables:

- Pos: Tupla de nombre enters (x,y) que conté la posició de l'agent dins de l'entorn.
- Type: Enter que indica el tipus de l'agent.

Accions:

- Moure's aleatòriament a una posició buida de l'entorn si no està content.

3.- Descriviu com és el món d'aquests agents. Si canviem el paràmetre torus de la funció init() de la classe Schelling i li assignem el valor True, com estem modificant aquest món? Com afecta aquest canvi als agents?

El món d'aquest sistema és una graella quadrada. Quan activem el paràmetre torus, estem convertint el món en un Tor (Donut), és a dir fem que s'ajunti la vora dreta de la graella amb l'esquerra i la superior amb la inferior, així aconseguim que ara totes les posicions de l'entorn tinguin 8 posicions veïnes. Això fa que si posem un valor de homofilia elevat, en el cas de la graella els agents mai es concentraran a les vores, perquè no tindran prou veïns del seu mateix tipus. En el cas del món en forma de dònut, això no passarà.

4.- Jugueu amb els paràmetres que pot introduir l'usuari. Descriviu com afecten aquests paràmetres al sistema.

Paràmetres:

- Agent Density: És la proporció de posicions ocupades per agents. Com més baixa és aquesta densitat més opcions de moviment tenen els agents que no estan contents, però també tenen menys veïns.
- Fraction Minority: És la proporció d'agents blaus dins de la població total d'agents. Quan aquest paràmetre s'allunya de 0.5 per qualsevol dels dos costats, més desproporció hi ha entre el nombre d'agents d'un tipus i de l'altre.
- Homophily: És el nombre de veïns que un agent necessita per estar content. Com més elevat és aquest número, més costa d'aconseguir que un agent estigui content.



## Exercici 2

En aquest exercici haureu d'afegir al problema un tercer tipus d'agent. Aquests agents tindran les mateixes variables i accions que els altres, però els distingirem amb un color diferent.

Per simplificar el problema, farem que dins de la minoria escollida per l'usuari, sempre hi hagi la meitat dels agents d'aquest nou tipus.

Canvis en la funció `init()` de la classe `Schelling` a l'arxiu `model.py`:

```
for cell in self.grid.coord_iter():
    x = cell[1]
    y = cell[2]
    if self.random.random() < self.density:
        if self.random.random() < self.minority_pc/2:
            agent_type = 2
        elif (self.random.random() >= self.minority_pc/2) & (self.random.random() < self.minority_pc):
            agent_type = 1
        else:
            agent_type = 0

        agent = SchellingAgent((x, y), self, agent_type)
        self.grid.position_agent(agent, (x, y))
        self.schedule.add(agent)
```

Canvis en la funció `ascii_agent()` de la classe `SchellingTextVisualization` a l'arxiu `server.py`:

```
@staticmethod
def ascii_agent(a):
    """
    Minority agents are X, Majority are O.
    """
    if a.type == 0:
        return 'O'
    if a.type == 1:
        return 'X'
    if a.type == 2:
        return 'Y'
```



Canvis en la funció `schelling_draw()` de la classe `HappyElement` a l'arxiu `server.py`:

```
def schelling_draw(agent):
    """
    Portrayal Method for canvas
    """
    if agent is None:
        return
    portrayal = {"Shape": "circle", "r": 0.5, "Filled": "true", "Layer": 0}

    if agent.type == 0:
        portrayal["Color"] = ["#FF0000", "#FF9999"]
        portrayal["stroke_color"] = "#00FF00"
    elif agent.type == 1:
        portrayal["Color"] = ["#0000FF", "#9999FF"]
        portrayal["stroke_color"] = "#000000"
    else:
        portrayal["Color"] = ["#FF00FF", "#FF99FF"]
        portrayal["stroke_color"] = "#4B0082"
    return portrayal
```

Torneu a jugar ara amb els paràmetres de l'usuari. Doneu un valor de 0.65 al paràmetre de Fraction minority, un valor de 0.7 al paràmetre d'Agent density i un valor de 4 al paràmetre d'Homophily. Si executeu el model uns quants cops, com acostuma a evolucionar el sistema després de 30 passes?

Els agents dels diferents colors s'han anat agrupant i després de només 30 passes ja s'han generat uns conjunts força diferenciats. D'altra banda el número d'agents contents no ha parat de créixer, però després de 30 passes aquest creixement ja és moderat (segueix la forma d'una funció logarítmica).

### Exercici 3

En aquest exercici afegireu una nova regla al problema: Quan un agent es trobi rodejat per més de 4 agents d'un altre color, aquest agent canviarà el seu tipus per el de la majoria dels seus veïns.

Canvis en la funció `step()` de la classe `SchellingAgent` a l'arxiu `model.py`:



```
def step(self):
    similar = 0
    color_dict={0:0,1:0,2:0}
    for neighbor in self.model.grid.neighbor_iter(self.pos):
        if neighbor.type == self.type:
            similar += 1
        color_dict[neighbor.type]=color_dict[neighbor.type]+1
    max_value,max_key=max(zip(color_dict.values(), color_dict.keys()))

    # If unhappy, move:
    if similar < self.model.homophily:
        if max_value > 4:
            self.type = max_key
        else:
            self.model.grid.move_to_empty(self)
    else:
        self.model.happy += 1
```

Torneu a jugar ara amb els paràmetres de l'usuari. Doneu un valor de 0.65 al paràmetre de Fraction minority, un valor de 0.7 al paràmetre d'Agent density i un valor de 5 al paràmetre d'Homophily. Si executeu el model uns quants cops, com acostuma a evolucionar el sistema?

Un dels tipus d'agent s'acaba menjant als altres dos. El número d'agents contents manté una tònica creixent.

#### Exercici 4

En aquest exercici se us demana que descriviu com aplicàrieu un sistema multi-agent a un problema d'actualitat on hi hagi la necessitat de que els agents es comuniquin entre ells. Algunes idees de problemes són:

- Vehicles autònoms
- Internet of Things (IoT)
- Minería de Bitcoins
- Nanorobots en tractaments mèdics

Però sentiu-vos lliures d'escollir qualsevol altre tema que trobeu més interessant.

Per al problema que hagueu escollit:

1.- Feu una breu introducció del problema

Es valorarà que l'estudiant faci una breu introducció del problema.

Per exemple: Tenim una ciutat on tots els vehicles són autònoms. Aquests vehicles estan equipats amb sensors, per poder reaccionar a diferents estats que rebin de l'entorn. També necessiten compartir informació, per evitar que hi hagi embussos en determinades zones o que hi hagi accidents en les cruïlles.



2.- Definiu el tipus d'agents que utilitzaríeu i quines dades recollirien.

Es valorarà que l'estudiant justifiqui el tipus d'agents que utilitzaria en aquest sistema.

Per exemple: Els agents d'aquest sistema són agents col·laboratius de tipus deliberatiu. Tenim tants agents com vehicles existeixin. Cada agent té integrats sensors GPS per determinar la seva posició, infrarojos per detectar vianants, càmeres per detectar senyals de trànsit i semàfors i també un sensor de velocitat. A més tenen un model simbòlic explícit de l'entorn (mapa de la ciutat) i un motor d'inferència (un algoritme de cerca que té en compte les zones amb més congestió).

3.- Descriviu el tipus d'accions que podrien realitzar aquests agents

Es valorarà que l'estudiant descrigui les accions d'aquests agents.

Per exemple: Aquests agents poden realitzar diverses accions depenent de les dades rebudes pels seus sensors. Frenar quan detecten un vianant, un semàfor en vermell o una senyal d'Stop, reduir o augmentar la velocitat quan detecten un senyal de trànsit de velocitat, etc.

4.- Discutiu quina tècnica de comunicació utilitzarien els agents i quina informació es passarien els agents entre ells.

Es valorarà que l'estudiant descrigui una tècnica de comunicació i la informació que compartirien els agents.

Per exemple: Podríem utilitzar un sistema de pissarra, on tots els vehicles enviessin a una plataforma central informació sobre la seva posició. Tots els agents coneixerien la posició dels altres agents i així, a partir d'aquesta informació, podrien calcular la forma òptima per arribar al seu destí en el menor temps possible, evitant les vies amb més congestió.

Els criteris de correcció de la PAC invaliden una A si tots els processos no estan ben justificats.

**Nota: Propietat intel·lectual**

Sovint és inevitable, en produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-ho en el marc d'una pràctica dels estudis d'Enginyeria Informàtica, sempre i això es documenti clarament i no suposi plagí en la pràctica.

Per tant, en presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en què es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU, GPL ...). L'estudiant haurà d'assegurar-se que la llicència que sigui no impedeix específicament seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida pel copyright.

Hauran, a més, adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.