# **Apuntes – Módulo 8**

## **Curso: Modeling and Simulation of Natural Processes**

**Universidad de Ginebra - Prof. Bastien Chopard et al.** **Semana 8 – Modelos Basados en Agentes**

## **1. Motivación y ejemplos fundamentales**

### **Observaciones naturales**

* **Comportamiento emergente**: Entidades individuales simples → comportamiento colectivo complejo
* **Casos de estudio principales**:
  + Hormigas formando montones de cadáveres (Messor sanctus)
  + Bacterias navegando hacia fuentes de azúcar (quimiotaxia)
  + Comerciantes en bolsas de valores

### **Pregunta central**

¿Cómo modelar entidades básicas individuales para observar comportamiento emergente global?

**Aplicaciones adicionales**:

* Simulación peatonal (evacuación de edificios)
* Propagación de epidemias
* Modelado ecológico (especies invasoras)
* Sistemas de optimización

## **2. Fundamentos: ¿Qué son los agentes?**

### **Definición básica**

* Entidad lógica autónoma y descentralizada
* Concepto de la Inteligencia Artificial
* Interactúa con el entorno (que puede incluir otros agentes)

### **Esquema básico**

ENTORNO ↔ AGENTE

↑ ↓

PERCEPCIÓN → ACCIÓN

### **Tipos de agentes**

#### **1. Agente reflejo simple**

* Sin memoria (apátrida)
* Comportamiento = función: PERCEPCIÓN → ACCIÓN
* Basado en reglas (puede ser estocástico)
* Percepción limitada del entorno

#### **2. Agente inteligente**

* Con memoria/estado interno
* Comportamiento: PERCEPCIÓN + ESTADO → ACCIÓN + NUEVO\_ESTADO
* Capacidad de aprendizaje
* Comportamiento más rico y adaptativo

### **Ejemplo práctico: Agente comercial**

def comportamiento(precio, estado):

ultimo\_precio, dinero, acciones = estado

if precio > ultimo\_precio and (precio - ultimo\_precio) > umbral:

cantidad = floor(acciones \* constante\_venta)

return VENDER(cantidad, precio)

elif precio < ultimo\_precio and (ultimo\_precio - precio) > umbral:

cantidad = floor((dinero \* constante\_compra) / precio)

return COMPRAR(cantidad, precio)

else:

return NO\_HACER\_NADA

## **3. Sistemas multiagente**

### **Características clave**

* Múltiples agentes en el mismo entorno
* Pueden ser idénticos, similares o diferentes
* Interacción directa o a través del entorno
* Cada agente ve a los demás como "caja negra"

### **Propiedades espaciales**

#### **Localización de agentes**

* **Dominio espacial**: 2D, 3D, o grafos
* **Movilidad**: los agentes pueden moverse
* **Topología**: afecta percepción e interacción

#### **Conciencia del entorno**

* **Radio de percepción**: área circular alrededor del agente
* **Línea de visión**: percepción direccional
* **Influencia en la red de interacción**

#### **Topología de interacción**

* **Completa**: todos interactúan con todos
* **Por contacto**: solo agentes adyacentes
* **Por conciencia espacial**: basada en percepción

## **4. Implementación computacional**

### **Enfoque orientado a objetos**

* **Clase** = tipo de agente
* **Instancia** = agente individual
* **Miembros privados** = estado interno
* **Métodos públicos** = comportamiento

### **Esquemas de actualización**

#### **Asíncrono**

while t < t\_max:

for agente in agentes:

percepcion = calcular\_percepcion(agente)

accion = agente.comportamiento(percepcion)

actualizar\_entorno(accion)

#### **Síncrono**

while t < t\_max:

percepciones = calcular\_todas\_percepciones()

acciones = todos\_comportamientos(percepciones)

actualizar\_entorno(acciones)

### **Enfoques espaciales**

#### **Lagrangiano**

* Cada agente conoce su posición exacta
* Flexibilidad espacial
* Complejidad O(n²) para interacciones locales
* Optimización con k-d trees: O(n log n)

#### **Euleriano**

* Grilla regular de celdas
* Agentes en listas por celda
* Ventajas: cálculo eficiente, paralelismo
* Desventajas: pérdida de precisión espacial

## **5. Caso de estudio: Agrupación de cadáveres de hormigas**

### **Fenómeno observado**

* Hormigas vivas agrupan cadáveres en montones
* Proceso emergente sin coordinación central
* Experimento: 8 horas → montón estructurado

### **Modelo de Deneubourg (1991)**

#### **Configuración**

* Grilla regular, 4 direcciones
* Caminata aleatoria sobre cadáveres
* Actualización asíncrona

#### **Comportamiento**

* **Recoger**: probabilidad Pp si cadáver aislado/cluster pequeño
* **Depositar**: probabilidad Pd si cluster grande
* **Memoria**: M de tamaño n (estados de últimas n celdas visitadas)

#### **Cálculo de Probabilidades**

f = Σ M(i) // suma de memoria

Pp = (k1/(k1 + f))²

Pd = (f/(f + k2))²

### **Modelo Universidad de Ginebra (2000)**

#### **Configuración**

* Grilla regular, 8 direcciones
* Caminata aleatoria con difusión alta
* Actualización asíncrona

#### **Comportamiento Simplificado**

* **Evitar**: cadáveres, otras hormigas, paredes
* **Regla simple**: hormiga sin carga recoge cadáver encontrado
* **Depositar**: hormiga cargada suelta al encontrar otro cadáver

#### **Mecanismo**

* Probabilidades iguales para remover/agregar cadáveres
* Fluctuaciones estadísticas → solo un cluster sobrevive
* No requiere "inteligencia" de las hormigas

### **Resultados comparativos**

* **Deneubourg**: convergencia ~10x más rápida
* **Ginebra**: más simple, sin memoria
* **Ambos**: comportamiento individual, no colectivo
* **Escalabilidad**: N(t) = f(Mt) - lineal con número de hormigas

## **6. Caso de estudio: Quimiotaxia Bacteriana**

### **Fenómeno observado**

* Bacterias nadan hacia fuentes de azúcar
* Imposible percibir gradientes espaciales directamente
* Movimiento aparentemente aleatorio → comportamiento dirigido

### **Mecanismo de movimiento**

* **Dos modos**:
  + Correr: movimiento en línea recta
  + Dar tumbos: cambio de dirección aleatoria
* **Flagelos**: rotación contraria a agujas del reloj → movimiento lineal
* **Turbulencia**: rotación en sentido horario → tumbo

### **Modelo computacional**

#### **Configuración**

* Grilla euleriana 2D
* Cada celda (x,y) contiene:
  + Lista de bacterias
  + Concentración de nutriente ρx,y

#### **Estado del agente**

* **di**: última dirección tomada (N, S, E, W)
* **mi**: última concentración percibida

#### **Comportamiento**

def comportamiento(rho, m\_i, d\_i):

if rho <= m\_i:

p = p\_d # probabilidad alta de tumbo

else:

p = p\_i # probabilidad baja de tumbo

if random() <= p:

return rho, direccion\_aleatoria()

else:

return rho, d\_i

#### **Parámetros**

* **pi**: probabilidad de tumbo cuando concentración aumenta
* **pd**: probabilidad de tumbo cuando concentración disminuye
* **Condición**: pd > pi

### **Mecanismo molecular**

* Bacterias recuerdan concentración anterior
* Comparación temporal en lugar de gradiente espacial
* Estrategia: continuar si mejora, cambiar si empeora

## **7. Consideraciones generales y conclusiones**

### **Ventajas de los modelos basados en agentes**

* **Intuitividad**: modelado directo de entidades observables
* **Emergencia**: comportamiento complejo desde reglas simples
* **Flexibilidad**: diferentes tipos de agentes en mismo sistema
* **Escalabilidad**: paralelización natural

### **Desafíos de implementación**

* **Complejidad computacional**: O(n²) para interacciones
* **Validación**: dificultad para validar comportamiento emergente
* **Parámetros**: muchos parámetros libres a calibrar
* **Tiempo**: usualmente tiempo continuo discretizado

### **Aplicaciones más allá del modelado**

* **Optimización**: algoritmos de enjambre
* **Seguridad de redes**: agentes móviles de monitoreo
* **Videojuegos**: NPCs con comportamiento autónomo
* **Robótica**: sistemas de robots colaborativos

### **Lecciones aprendidas**

1. **Simplicidad vs. Realismo**: modelos simples pueden explicar fenómenos complejos
2. **Emergencia**: comportamiento colectivo ≠ comportamiento individual
3. **Validación empírica**: importancia de contrastar con datos experimentales
4. **Escalabilidad**: efectos lineales vs. no lineales en sistemas multiagente

### **Implementación de modelos basados en agentes**

* Detalles de programación orientada a objetos
* Optimizaciones espaciales avanzadas
* Frameworks y herramientas existentes
* Gestión de memoria y rendimiento

### **Validación y verificación**

* Métodos de validación de modelos
* Análisis de sensibilidad de parámetros
* Comparación con datos experimentales
* Métricas de evaluación

### **Aplicaciones avanzadas**

* Modelos epidemiológicos
* Simulación de tráfico y evacuación
* Ecosistemas y biodiversidad
* Mercados financieros

En este módulo se representa un enfoque fundamental para entender sistemas complejos desde una perspectiva bottom-up, donde el comportamiento global emerge de interacciones locales simples.