Seminario de Ciencias de la Computación "A"
Modelación y simulación computacional basada en agentes 2021-I

Práctica 2
"Modelos Basados en Agentes"

1. Modelo de segregación de Schelling

El modelo propuesto originalmente por Thomas Schelling[1] consiste en dos grupos de agentes, por ejemplo, rojos y verdes, que localmente tratan de satisfacer la necesidad de estar con los de su mismo grupo. De manera general este comportamiento es establecido con un parámetro conocido como porcentaje de *similitud-requerida* o nivel de tolerancia.

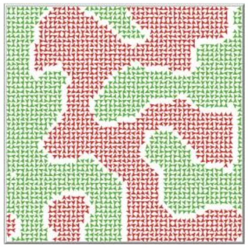
Los agentes toman una decisión a partir de la información que tienen en su vecindad. Si el agente satisface las condiciones del entorno entonces se queda en su posición actual, de lo contrario, se mueve a una posición vacía. Esta dinámica local genera como resultado la formación de cúmulos de agentes del mismo tipo, es decir, hay segregación.

Definición del sistema (entorno): el sistema se compone de una retícula de *n* x *n* donde *n* se establece usualmente entre 50 y 100. Cada celda con posición (i,j) alberga a un agente rojo o verde. El sistema tiene un parámetro de *densidad poblacional*, usualmente se establece en 90% (es decir, 10% de las celdas quedan vacías). La mitad de la población se inicializa como rojos y la otra como verdes. Cada agente toma una celda de manera aleatoria.

Dinámica (reglas): cada agente en la posición (i,j) se "muda" a un lugar vacío si en su vecindad de Moore (8-vecinos) no cumple con el porcentaje de similitud requerida.

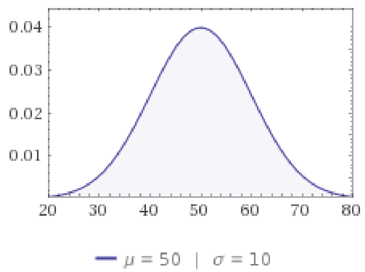
Ejercicios:

1. Implemente el modelo de segregación de Schelling original, (pueden usar el código de la biblioteca de modelos de NetLogo, si lo programan en otro lenguaje de programación tienen un punto extra)
2. Establezca el tamaño de retícula como n=50, con densidad poblacional del 90% ¿Qué valor del parámetro de similitud es el límite máximo para formar dinámicas de segregación? A este valor le llamaremos Smax.
3. Una propuesta de medida para detectar convergencia es cuando los agentes ya no cambian de posición. Por ejemplo:



El sistema en el tiempo t es igual en el tiempo (t+1).

Cuándo el parámetro de similitud es igual a Smax, ¿cuál es el tiempo en el que el sistema converge? De manera general, forme una gráfica parámetro-similitud vs tiempo-de-convergencia. ¿Cómo crece/decrece el tiempo de convergencia en función del parámetro de similitud? ¿lineal, logarítmico, exponencial? Cuando el sistema no converja (probar un tiempo suficientemente grande) dejar de graficar.
4. **Extensión del modelo**. Establezca el parámetro de similitud como un atributo de los agentes. Inicialice la similitud requerida del agente i-ésimo a partir de una distribución normal con media 50 y desviación estándar 10. ¿Cómo cambian los patrones de segregación? ¿y con media = Smax y desviación estándar pequeña y grande?



Describa sus resultados y adjunte capturas de pantalla para dar soporte a la explicación. Hint: NetLogo tiene definida una función generadora de números aleatorios con distribución normal:

random-normal *mean standard-deviation*

5. Modifique su programa previo para considerar tres tipos de agentes (rojos, verdes y azules). Inicialice cada grupo como 1/3 de la población y establezca de **manera global** el parámetro de similitud-requerida. ¿Se forman patrones de segregación? ¿Cuál es el valor del umbral Smax? Adjunte capturas de pantalla y explique la dinámica.

---
[1] Schelling T.C. (1971). Dynamics models of segregation. *Journal of Mathematical Sociology*, Vol. 1, pp. 143-186.

6. Bajo su criterio que otros elementos de modelación se podrían definir en el modelo de Schelling para hacerlo más realista. Explique. ¿Qué otros análisis podrían implementar para explicar las dinámicas? Explique.

2. Termitas apiladoras

Este modelo fue propuesto por Mitchel Resnick[2] como una estrategia descentralizada para apilar astillas de madera (objetos) a través de simples reglas ejecutadas por termitas (agentes).

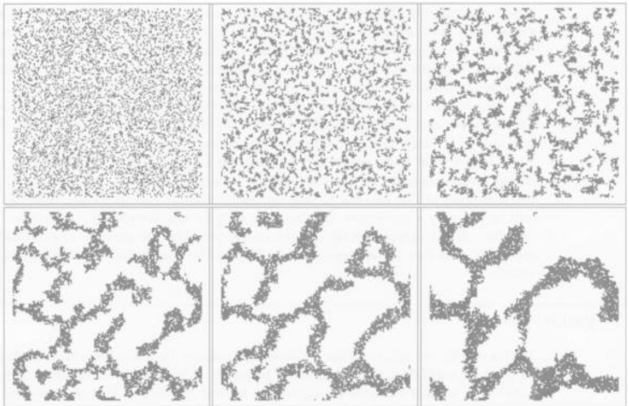
Definición del sistema (entorno): el sistema se compone de una retícula de n x n donde n = 100. Cada celda con posición (i,j) alberga una astilla de madera (color amarillo). El sistema tiene un parámetro de *número de termitas* y *densidad de astillas*.

Dinámica (reglas): Las termitas tienen dos reglas básicas.
A) Si la termita no está cargando nada y se encuentra una astilla de madera, la recoge.
B) Si está cargando una astilla de madera y se encuentra otra, suelta la astilla y continua su camino.

El movimiento de la termita es como un caminador aleatorio con una apertura de visión de -50 a 50 grados.

Ejercicios.

1. Implemente el modelo de termitas apiladoras, (pueden usar el código de la biblioteca de modelos de NetLogo, si lo programan en otro lenguaje de programación tienen un punto extra).
2. Implemente una gráfica donde se observe el comportamiento del sistema en función del tiempo, por ejemplo, el número de cúmulos en función el tiempo, el promedio del tamaño del cúmulo en función del tiempo, o el número de termitas que están cargando astillas. Si encuentra otra forma para obtener información, grafique y argumente porque es adecuada.
3. Extender el modelo considerando dos tipos de astillas de madera (por ejemplo, amarillas y cafés). La termina deja y recoge la astilla a partir del color del cúmulo. ¿cuántas pilas de astillas quedan al final? Muestre la evolución del sistema a partir de captura de pantallas del sistema.
4. En la regla original, la termita suelta la astilla si encuentra otra astilla del mismo color y sigue su camino. Implemente el siguiente comportamiento: una vez que suelta la astilla, la termita "salta" a otra posición de manera aleatoria. Este pequeño cambio genera la estructura de la siguiente Figura[3]. Capture la pantalla de estados finales y explique el comportamiento. ¿Esta estructura se deduce a partir de las reglas locales?



[2] Resnick, M. (1977). *Turtles, termites, and traffic jams. Explorations in Massively Parallel Microworlds*. MIT Press.

Parte 3. Hormigas en el borde del caos

En el artículo "Orden y caos en la organización social de las hormigas" de Octavio Miramontes, el autor define un modelo computacional para estudiar el comportamiento de las hormigas a partir de la cantidad de agentes que se encuentran en el ambiente. Se define un mecanismo para activar a las hormigas y las interacciones entre ellas.

Con la información que se encuentra en el artículo, implemente un modelo basado en agentes donde consideré características como: la función de activación (tipo tangente hiperbólica), una vecindad de Moore, movimiento aleatorio de las hormigas, tamaño de retícula y variación de la población. Replique algunos de los resultados más importantes como los patrones colectivos de activación que dependen de la variación de densidades de hormigas sobre el espacio.

1. Crear las series de tiempo de activación para distintas densidades de la población de hormigas. Específicamente, contar las hormigas que se encuentran en estado activo en el tiempo t, y graficar durante la evolución (600 tiempos).
2. Realizar las gráficas para distintas densidades de la población (de 5 a 10 gráficas), de tal manera que las puedan comparar y ver la transición orden-desorden (ver Figura 1), mostrar que con densidades

[3] Flake G.W. (1999). *The Computational Beauty of Nature. Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*. The MIT Press, pp. 263.

aproximadamente del 20% el sistema exhibe una transición de fase explicar detalladamente que sucede.

3. **(Optativo)** Usar algún algoritmo de compresión (tar, gzip, bzip2, tgz, rar …) para compactar las series de tiempo, medir la tasa de compresión para cada una de las series (variando densidades desde 1% hasta 100%) y graficar, los resultados deben ser compatibles con la Figura 2.

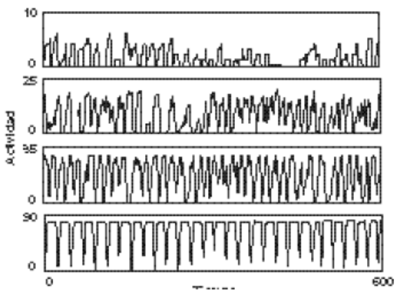


Figura 1

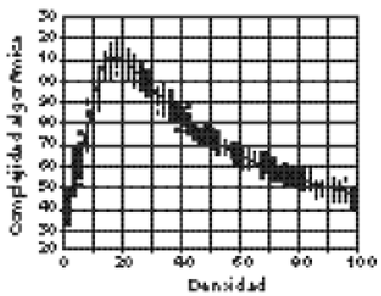


Figura 2.

4. Tema para proyecto final

A lo largo del curso se han revisado varios sistemas modelados con Autómatas Celulares y Modelación Basada en Agentes, por ejemplo: a) regeneración de bosques b) dinámica del tráfico vehicular, b) dinámica de incendios forestales, c) procesos infecciosos (epidemias) d) Sistema de reacción-difusión e) “Boids” f) Modelo de la pila de arena g) Modelo de Rumores, g) Modelo de rebelión, h) insectos sociales, etc.

Para realizar el proyecto final deberán escoger un tema de su interés sustentando en publicaciones académicas (libros o artículos) para cumplir el siguiente objetivo:

1. Replicar algunos de los resultados de algún artículo científico donde se use modelación basada en agentes.
2. Extender el modelo a través de nuevas reglas, condiciones de entorno.
3. Proponer un nuevo modelo.

Para guiarse en este proceso pueden considerar los siguientes puntos:

1. Revisar los artículos de Dropbox o realizar búsquedas en internet con el tema de su interés. **Subir los artículos al Dropbox para compartirlos con la clase.**
2. Estudiar y analizar el sistema, consultar apuntes y referencias dadas en clase. Revisar las características del modelo: posibles estados de las celdas, reglas de evolución, inicialización del sistema, condiciones de frontera, parámetros globales, parámetros locales, etc.
3. Observar en que lenguaje de programación está implementado.
4. Observar la inicialización del sistema. Identificar los parámetros del sistema. ¿Qué pasa cuando se cambian los valores de los parámetros?
5. Observar el comportamiento global del sistema, ¿tiene alguna propiedad emergente?
6. ¿Representa verdaderamente lo que pasa en la realidad? el modelo explica el fenómeno o predice algún tipo de comportamiento.

Opcional. Loop de Langton.

Christopher Langton mostró la capacidad de autoreproducción de los autómatas celulares a partir de los modelos de Von Neumann y Codd. El loop de Langton es un modelo simple el cual satisface los criterios de autoreproducción. Esta estructura logra su simplicidad almacenando su descripción en un "loop" dinámico.

El modelo se desarrolla en un autómata celular bidimensional con 8 posibles estados en sus celdas y una vecindad de Neumann

Condición inicial del sistema

```
  2 2 2 2 2 2 2 2
2 1 7 0 1 4 0 1 4 2
2 0 2 2 2 2 2 2 0 2
2 7 2         2 1 2
2 1 2         2 1 2
2 0 2         2 1 2
2 7 2         2 1 2
2 1 2 2 2 2 2 2 1 2 2 2 2 2
2 0 7 1 0 7 1 0 7 1 1 1 1 1 2
  2 2 2 2 2 2 2 2 2 2 2 2 2
```

Descripción general de los estados

Cada estado identifica de forma general alguna parte del loop de Langton.

a) El estado "1" indican las celdas del "núcleo" o canal de datos.
b) El estado "2" son las celdas "coraza", el recubrimiento del canal de datos.
c) Los estados "4", "5", "6" y "7" seguidas por un "0" son paquetes de información que viajan por el canal de datos, dependiendo de los valores asignados en la tabla de transiciones será la función que realicen.
d) El estado "3" en combinación con sus vecinos indican la apertura o cierre del canal de datos (no se utiliza en la condición inicial).

Tabla de transiciones

En la siguiente tabla se muestra los valores de la transición y la posición de cada valor de la celda dependiendo de su ubicación, "C" (center), "T" (top), "R" (right), "B" (bottom), "L" (left). Hay que considerar las posibles rotaciones sin tomar en cuenta "C", es decir, CTRBL = CRBLT = CBLTR = CLTRB.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
00000->0	02527->1	11322->1	20242->2	30102->1
00001->2	10001->1	12224->4	20245->2	30122->0
00002->0	10006->1	12227->7	20252->0	30251->1
00003->0	10007->7	12243->4	20255->2	40112->0
00005->0	10011->1	12254->7	20262->2	40122->0
00006->3	10012->1	12324->4	20272->2	40125->0
00007->1	10021->1	12327->7	20312->2	40212->0
00011->2	10024->4	12425->5	20321->6	40222->1
00012->2	10027->7	12426->7	20322->6	40232->6
00013->2	10051->1	12527->5	20342->2	40252->0
00021->2	10101->1	20001->2	20422->2	40322->1
00022->0	10111->1	20002->2	20512->2	50002->2
00023->0	10124->4	20004->2	20521->2	50021->5
00026->2	10127->7	20007->1	20522->2	50022->5
00027->2	10202->6	20012->2	20552->1	50023->2
00032->0	10212->1	20015->2	20572->5	50027->2
00052->5	10221->1	20021->2	20622->2	50052->0
00062->2	10224->4	20022->2	20672->2	50202->2
00072->2	10226->3	20023->2	20712->2	50212->2
00102->2	10227->7	20024->2	20722->2	50215->2
00112->0	10232->7	20025->0	20742->2	50222->0
00202->0	10242->4	20026->2	20772->2	50224->4
00203->0	10262->6	20027->2	21122->2	50272->2
00205->0	10264->4	20032->6	21126->1	51212->2
00212->5	10267->7	20042->3	21222->2	51222->0
00222->0	10271->0	20051->7	21224->2	51242->2
00232->2	10272->7	20052->2	21226->2	51272->2
00522->2	10542->7	20057->5	21227->2	60001->1
01232->1	11112->1	20072->2	21422->2	60002->1
01242->1	11122->1	20102->2	21522->2	60212->0
01252->5	11124->4	20112->2	21622->2	61212->5
01262->1	11125->1	20122->2	21722->2	61213->1
01272->1	11126->1	20142->2	22227->2	61222->5
01275->1	11127->7	20172->2	22244->2	70007->7
01422->1	11152->2	20202->2	22246->2	70112->0
01432->1	11212->1	20203->2	22276->2	70122->0
01442->1	11222->1	20205->2	22277->2	70125->0
01472->1	11224->4	20207->3	30001->3	70212->0
01625->1	11225->1	20212->2	30002->2	70222->1
01722->1	11227->7	20215->2	30004->1	70225->1
01725->5	11232->1	20221->2	30007->6	70232->1
01752->1	11242->4	20222->2	30012->3	70252->5
01762->1	11262->1	20227->2	30042->1	70272->0
01772->1	11272->7	20232->1	30062->2	

```
  T
L C R      ==>      I
  B
```

Ejercicios

1) Implemente en el lenguaje de programación de su preferencia el Loop de Langton. Utilice colores para representar los valores de los estados. El autómata debe tener por lo menos una longitud de 150x150 para observar adecuadamente el crecimiento de una colonia de loops.
2) Muestre la captura de pantalla en el tiempo 151. Aquí se cumplen dos generaciones del loop de Langton.
3) Muestre la captura de pantalla en la generación 7, ¿cuántos loops hay "muertos"? ¿cuántos "moribundos"? y ¿cuántos en fase de reproducción?
4) Puede deducir una fórmula para el crecimiento de la colonia de los loops.

Lineamientos para la entrega

Entregar un empaquetado (.tgz, .zip, etc.) con la siguiente estructura:

```
Apellido
   |--------- PracticaN
                |---------- fuentes
                |---------- readme.txt
                |---------- solución a ejercicios.pdf
```

Fuentes: código del programa, ejecutables, etc.

Readme.txt: explicar brevemente como se ejecuta el programa, y la forma en que podemos cambiar parámetros.

SolucionEjercicios.pdf: En este archivo incluir todas las respuestas de la práctica junto con las gráficas y comentarios extras.

Enviar la práctica al mail gcarreon@unam.mx y vanchristoph3r@gmail.com

Fecha de entrega: 6 de diciembre antes de las 24:00 hrs.

Fecha para presentar el tema de su proyecto final: lunes 7 y martes 8 de diciembre. En la clase platicarán sobre el tema (10 min máximo). Realizar una o dos diapositivas con la siguiente información: tema, objetivo, descripción del modelo y ficha bibliográfica.

Contacto
Gustavo Carreón gcarreon@unam.mx
Christopher Chávez vanchristoph3r@gmail.com

Lista del curso evolutivo@listas.iiec.unam.mx