

Práctica 6

Sistema multiagente

Introducción

Un sistema multiagente, como su nombre lo dice, está compuesto de diversos agentes; cada agente tiene comportamiento autónomo y dinámico e interactúa con un ambiente y otros agentes, causando un cambio en el ambiente en el que está expuesto.

En esta práctica, se toma en cuenta el sistema multiagente para simular el progreso de un agente infectado al interactuar con agentes susceptibles a infectarse, con la probabilidad de recuperarse y no volver a contraer la enfermedad, sin olvidar realizar esta simulación en paralelo, para hacer más eficiente dicha simulación basándonos en el modelo *SIR*, que es un modelo muy elemental para explicar la evolución de alguna enfermedad infecciosa, creada por algún virus o bacteria. En este caso, se dividen a los agentes en: * **S**: Agentes susceptibles, los que tienen riesgo de contraer la enfermedad. * **I**: Agentes infectados, aquellos que contrajeron la enfermedad y pueden contagiar a los susceptibles. * **R**: Agentes recuperados, que se recuperaron de la enfermedad y son inmunes a ella (en este caso).

Objetivos:

- Paralelizar la simulación para mejorar la eficiencia.
- Estudiar el efecto de la probabilidad de infección inicial en el porcentaje máximo de infectados.

Simulación

Para simular este sistema multiagente, se colocó el número de agentes que participarían en la simulación, la probabilidad inicial de infección, la probabilidad de recuperación y la velocidad con la que los agentes se mueven en la gráfica. Cuando un agente alcance los límites visuales del espacio, se va a la primera

posición de su lado opuesto, por ejemplo, si está arriba se posiciona abajo y viceversa, y si está a la izquierda se posiciona a la derecha y viceversa.

La probabilidad de agentes infectados al inicio varía, ya que hay una función pseudoaleatoria, que genera una probabilidad menor de 0.05 para generar infectados al inicio, y asignar un estado para los agentes, de S , I o R .

La función que se ejecutará cada vez que vaya a cambiar de generación, o sea, el siguiente cuadro que se va a generar en los pasos de la simulación es la de *agentesPorTiempo*, la cual necesita estar paralelizada porque se ejecutará en cualquier momento en que se inicie la simulación. La función *agentesPorTiempo* tiene agregada funciones, primero, si el agente S tiene una interacción con un agente I , dependiendo de las coordenadas en que se encuentren, si chocan, hay una probabilidad de que el agente S se convierta en agente I . Después, si hay un agente S , entra a una condición con cierta probabilidad de recuperarse, si lo hace, su estado cambia a R . Al finalizar esta función, regresará los agentes guardados como si fuera una tabla, este proceso se estará efectuando 100 veces. Se añaden las líneas para hacer el proceso en paralelo, en donde la variable *agentesN* corresponde a la generación actual de los agentes.

```
clusterExport(cluster, "agentes")
agentesN <- foreach(i = 1:n, .combine=rbind) %dopar% agentesPorTiempo(i)
agentes <- agentesN
stopImplicitCluster()
```

Se usó un `rbind` para poder agregarle la información a la parte gráfica y poderla poner para que veamos el avance de este sistema multiagente.

Resultados

Al momento de paralelizar la simulación, hubo un cambio significativo en el rendimiento de la computadora, pero no solo eso, era más fácil realizar los procesos y que la máquina no tuviera que desgastarse mucho al momento de repetir ese proceso constantemente hasta que se termine.

Se cambió la velocidad para ver si tenía un efecto en los agentes, si tardarían en infectarse, y en 100 repeticiones, con la velocidad de 1/10, y la única variación fue que habían más agentes S al finalizar la simulación, comparado con otras velocidades.

También, el tiempo en el que es más probable que haya un agente infectado, es en el rango de 40-60 generaciones a lo largo del tiempo de la simulación.

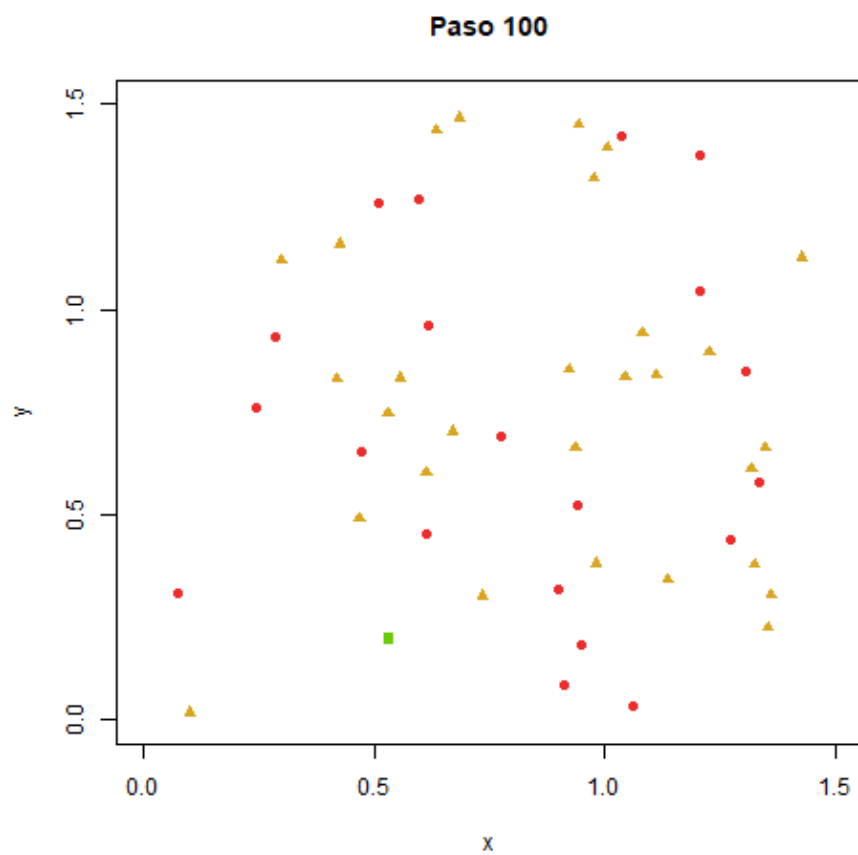


Figura 1: Diferencia de imágenes de diferentes velocidades. Velocidad 1/10

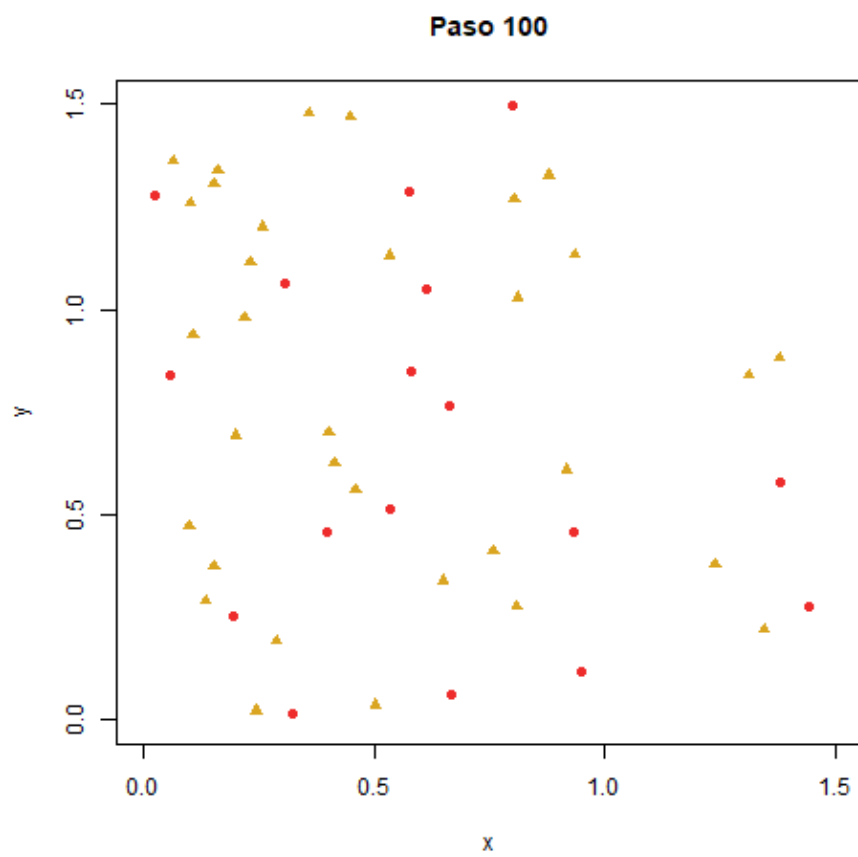


Figura 2: Diferencia de imágenes de diferentes velocidades. Velocidad $1/30$

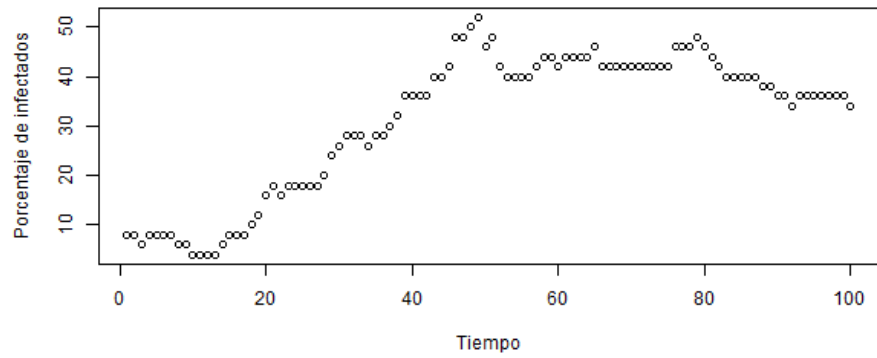


Figura 3: En el rango de 40-60, hay más porcentaje de infectados

Conclusión

La paralelización en métodos que se ejecutarán constantemente es muy recomendable, ya que la eficiencia de los procesos es buena, o sea, se tarda menos tiempo en hacer la simulación.